

UNIT - IV

4

Selenium Tool

4.1 : Introducing Selenium

Q.1 What is selenium ?

- Ans. :**
- Selenium is a popular framework for testing the user interface (UI) of a web application. It is an Open Source tool for automating browser-based applications.
 - Selenium is a suite of testing automation tools used for Web-Base applications: Selenium IDE, Selenium RC, Selenium WebDriver and Selenium Grid.
 - Selenium is a set of different software tools, each with a different approach to support test automation.
 - The tests can be written as HTML tables or coded in a number of popular programming languages and can be run directly in most modern Web browsers.
 - Selenium can be deployed on Windows, Linux, and Macintosh and many OS for mobile applications like iOS, Windows Mobile, and Android.

Q.2 How to decide use of testing tool for organization ?

- Ans. :**
- Management takes decision of introducing tools for software testing. But management committee members are not used tools. Their decision based on some success story, from sales person, from new letter etc. There are not verify the feasibility of the testing tool.

- After few weeks, when tester found that tools are not suitable for their work, they complained against tools. So management thinks their decision was wrong and waste of money and time.

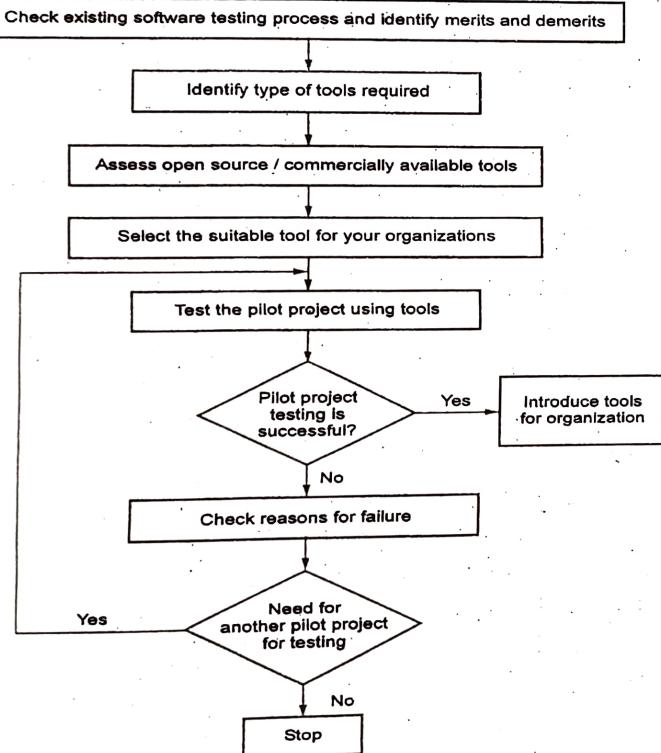


Fig. Q.2.1 Introducing tool in the organization process

- So it is necessary to check feasibility of tools and verify the necessity of tools by test engineer. Fig. Q.2.1 shows process of introducing tool in the organization.
- Take a review of current testing process and identify the problem faced by tester. Identify the total cost spending on testing process. Also check the customer feedback after submitting the software product. Check the overall process of testing and then decide about new system.
- Based on the project requirements, identify the tools required for your organization.
- Commercially available tools and open source tools are available in the market. Evaluate the both types of tools. Select one of tool and test the pilot project on the tool.
- Find out the rule and parameters before testing the project. If pilot project is successful, then introduce the tools for other project.
- If pilot project is not successful, then test the another project or discard the tools

Q.3 Explain benefit of testing tools.

Ans. :

- Relieve the testing bottleneck and achieve faster application time-to-market
- Reduce the money spent on testing
- Increase test coverage & reduce risk
- Configure and repeat your tests
- Re-assign your resources
- Deliver highly accurate tests and find defects earlier
- Ensure corporate compliance
- Ensure the scalability of your applications

- Test data creation, management, and security.

Q.4 Explain brief history of the Selenium project.

Ans. : • At first, Selenium was used only internally by ThoughtWorks employees. But that changed by the end of 2004 when the tool was open-sourced.

- Selenium first came to life in 2004 when Jason Huggins was testing an internal application at ThoughtWorks.
- Selenium core is born whose functionality underlies the Selenium RC (Remote Control) and selenium IDE tools.
- The limitation of having a JavaScript based automation engine and browser security restricted Selenium to specific functionality
- Google, who has been a long time user of Selenium, had a developer named Simon Stewart who developed WebDriver. This tool circumvented Selenium's JavaScript sandbox to allow it to communicate with the Browser and Operating System directly using native methods.
- In 2006 an engineer at Google named Simon Stewart started work on a project he called WebDriver.
- In 2008, the whole Selenium team decided to merge Selenium Webdriver with Selenium RC in order to form more powerful tool called Selenium 2.0.
- Selenium IDE, Selenium RC, Selenium WebDriver and Selenium Grid are the Selenium Projects.
- But in 2011, the projects were merged to form one web testing tool to rule them all. The combination of Selenium and WebDriver became Selenium 2.0, which debuted in July 2011.
- The new release paired the WebDriver APIs that are familiar to Selenium users today with the original Selenium feature set.
- Selenium had massive community and commercial support, but WebDriver was clearly the tool of the future.

- The joining of the two tools provided a common set of features for all users and brought some of the brightest minds in test automation under one roof.

Selenium 1.0 → (Selenium IDE + Selenium RC + Selenium Grid)

Selenium 2.0 (Selenium 1.0 + WebDriver)

(Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)

4.2 : Selenium's Tool Suite

Q.5 Explain Selenium's tool suite.

Ans. :

- It has four components,

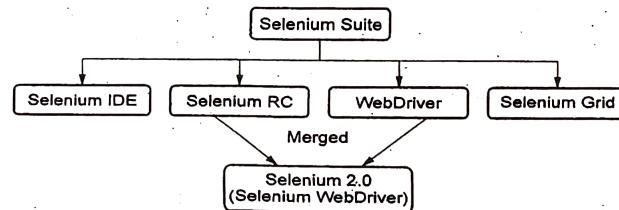


Fig. Q.5.1

1. Selenium IDE

- Rapid prototyping tool for building test scripts.
- It can be used by developers with little to no programming experience to write simple tests quickly
- It has a recording feature that records a user's live actions that can be exported in one of many programming languages.
- It does not provide iteration or conditional statements for test scripts.

- It can only run tests against FireFox.
- Developed tests can be run against other browsers, using a simple command-line interface.

2. Selenium RC server

- Selenium RC is the main Selenium project allowing user actions to be simulated in a browser like clicking a UI element, input data, etc.
- It executes the user commands in the browser by injecting Java script functions to the browser when the browser is loaded.

3. Selenium Webdriver

- Selenium WebDriver fits in the same role as RC did, and has incorporated the original 1.x bindings.
- It refers to both the language bindings and the implementations of the individual browser controlling code.
- This is commonly referred to as just "WebDriver" or sometimes as Selenium 2.

4. Selenium Grid

- Scales the Selenium RC solution for large test suites and test that must be run in multiple environments.
- Tests can be run in parallel with simultaneous execution (different tests on different remote machines)
- It allows for running your tests in a distributed test execution environment.
- Used to run your tests against multiple browsers, multiple versions of browser, and browsers running on different operating systems.
- It reduces the time it takes for the test suite to complete a test pass.

4.3 : Selenium-IDE

Q.6 Write short note on Selenium IDE

Ans. :

- Selenium IDE (Integrated Development Environment) is a prototyping tool for building test scripts.
- It is a Firefox plugin and provides an easy-to-use interface for developing automated tests.
- Selenium IDE has a recording feature, which records user actions as they are performed and then exports them as a reusable script in one of many programming languages that can be later executed.
- Selenium IDE is simply intended as a rapid prototyping tool.
- As a Firefox plugin, selenium IDE can be used to create test script prototype quickly and easily.
- It can record human tester actions as a script (in any programming language) while tester run the test case manually.
- The recorded test script can be executed later for regression test automatically. This tool can access browser's DOM (Document Object Module) elements use Javascript.
- Commonly Used Selenium Commands are as follows :

1. Open
2. click/clickAndWait
3. verifyTitle/assertTitle
4. verifyTextPresent
5. verifyElementPresent
6. verifyText
- 7.verifyTable
8. waitForPageToLoad
- 9.waitForElementPresent

- Using Firefox, first, download the IDE from the SeleniumHQ downloads page When downloading from Firefox, you'll be presented with the Window A.
- Select Install Now. The Firefox Add-ons window pops up, first showing a progress bar, and when the download is complete, displays the Window B.
- Restart Firefox. After Firefox reboots you will find the Selenium-IDE listed under the Firefox Tools menu.
- Fig. Q.6.1 shows Selenium IDE user interface.

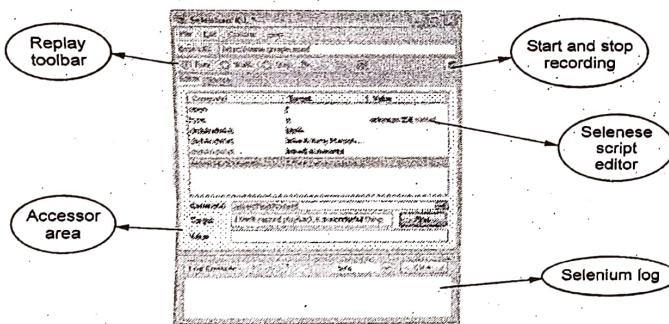


Fig. Q.6.1

Features :

- Easy to record and play
- Walk through tests
- Support debug and set breakpoints
- Auto complete for all common Selenium commands.

Q.7 Explain benefits and limitation of Selenium IDE.

Ans. : Benefits of Selenium IDE :

- Very easy to install and use.

- programming experience is required.
- Has built-in help feature and it shows the documentation on the selected or entered command.
- Helps in debugging by displaying the information and error messages.
- It allows us to set breakpoints, insert commands and comments wherever necessary.
- Provides a good support for extensions.
- Helps in exporting tests that can be executed in Selenium RC and WebDriver.

Limitations of Selenium IDE :

- As it comes as a Firefox plug-in, it doesn't support any browser other than Mozilla Firefox.
- It doesn't explicitly provide help to support alerts, pop-ups and navigations.
- It doesn't support listeners.
- No support for error handling and database testing.
- It cannot be used in the testing of iPhone and Android applications.
- Reading from external files and uploading files is not supported.
- No support for iterations and conditional operations.

Q.8 How to run Selenium IDE test case ?

Ans. :

- Run a Test Case : Click the Run button to run the currently displayed test case.
- Run a Test Suite : Click the Run All button to run all the test cases in the currently loaded test suite.
- Stop and Start : The Pause button can be used to stop the test case while it is running. The icon of this button then changes to indicate the Resume button. To continue click Resume.

4. Stop in the Middle : You can set a breakpoint in the test case to cause it to stop on a particular command. This is useful for debugging your test case. To set a breakpoint, select a command, right-click, and from the context menu select Toggle Breakpoint.
5. Start from the Middle : You can tell the IDE to begin running from a specific command in the middle of the test case. This also is used for debugging. To set a start point, select a command, right-click, and from the context menu select Set/Clear Start Point.
6. Run Any Single Command : Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly. This is also available from the context menu.

Q.9 How to record a Selenium test case ?

Ans. :

- Open Firefox that has the IDE installed
- Open the base URL of the application to record.
- Keep the application in a common base state.
- Go To Tools Selenium IDE and the IDE will be opened
- Now perform the operations on the application as you are testing the application.
- Once you are done with the recording click on the stop recording button and save the test case through the file menu. By default it will be saved as a selenese script (HTML format)

Q.10 Explain Menu bar of Selenium IDE.

Ans. : 1. Menu Bar :

- File menu has options for Test Case and Test Suite (suite of Test Cases). Using these you can add a new Test Case, open a Test

- Case, save a Test Case, export Test Case in a language of your choice.
- You can also open the recent Test Case. All these options are also available for Test Suite.
 - The Edit menu allows copy, paste, delete, undo, and select all operations for editing the commands in your test case.
 - The Options menu allows the changing of settings.
 - You can set the timeout value for certain commands, add user-defined user extensions to the base set of Selenium commands, and specify the format (language) used when saving your test cases.
 - The Help menu is the standard Firefox Help menu; only one item on this menu-UI-Element Documentation-pertains to Selenium-IDE.

2. Toolbar

- The toolbar contains buttons for controlling the execution of your test cases, including a step feature for debugging your test cases.
- The right-most button, the one with the red-dot, is the record button.
- Speed Control : Controls how fast your test case runs.
- Run All : Runs the entire test suite when a test suite with multiple test cases is loaded.
- Run : Runs the currently selected test. When only a single test is loaded this button and the Run All button have the same effect.
- Pause/Resume : Allows stopping and re-starting of a running test case.
- Step : Allows you to "step" through a test case by running it one command at a time. Use for debugging test cases.
- TestRunner Mode : Allows you to run the test case in a browser loaded with the Selenium-Core TestRunner. The TestRunner is

not commonly used now and is likely to be deprecated. This button is for evaluating test cases for backwards compatibility with the TestRunner. Most users will probably not need this button.

4.4 : Selenium RC

Q.11 What is Selenium RC ? Explain software required for installing Selenium RC.

Ans. : In Selenium RC, the Selenium Server launches and kills browsers, interprets and runs the Seleneese commands passed from the test program, and acts as an HTTP proxy, intercepting and verifying HTTP messages passed between the browser and the AUT.

- If your AUT is behind an HTTP proxy which requires authentication then you should configure http.proxyHost, http.proxyPort, http.proxyUser and http.proxyPassword.
 - You can run Seleneese html files directly within the Selenium Server by passing the html file to the server's command line.
 - Client libraries which provide the interface between each programming language and the Selenium RC Server.
 - The primary task for using Selenium RC is to convert your Seleneese into a programming language.
 - A solution to cross browser testing. A server, written in Java and so available on all the platforms.
 - Acts as a proxy for web requests from them. Client libraries for many popular languages.
 - Bundles Selenium Core and automatically loads into the browser.
- Fig. Q.11.1 shows architecture diagram.

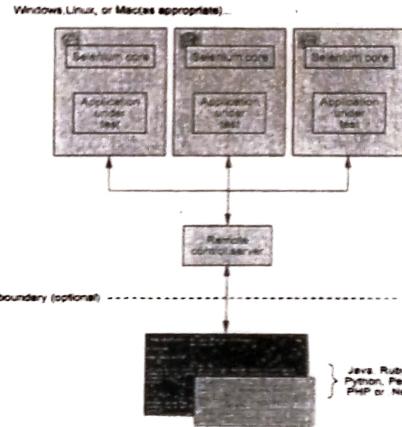


Fig. Q.11.1

1. Tester write test case script with supported programming language API.
2. When test script is running, it open a web browser and injects selenium core which is just a suite of Javascripts, then use these Javascripts to execute command.
3. Test script send command to RC server.
4. RC server receives these command and trigger selenium core to execute the commands and interact with browser page web elements.

4.5 : Selenium Webdriver and Selenium Grid

Q.12 What is Selenium Webdriver ? What are the reasons to use the Selenium-Server with Selenium-WebDriver ?

Ans. : • WebDriver is designed to providing an simpler, more concise programming interface.

- Selenium-WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded.
- WebDriver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems.
- It makes direct calls to the browser using each browser's native support for automation.
- WebDriver is a tool for automating web application testing, and in particular to verify that they work as expected.
- Reasons to use the Selenium-Server with Selenium-WebDriver:
 1. Using Selenium-Grid to distribute tests over multiple machines or virtual machines.
 2. To connect to a remote machine that has a particular browser version that is not on current machine.
 3. If users are not using the Java bindings (i.e. Python, C#, or Ruby) and would like to use HTMLUnit Driver

Q.13 Write short note on Selenium Grid.

- Ans. :
- Selenium-Grid allows tester run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and OS.
 - Essentially, Selenium-Grid support distributed test execution. It allows for running tests in a distributed test execution environment.
 - Scales the Selenium RC solution for large test suites and test that must be run in multiple environments.
 - It allows for running tests in a distributed test execution environment.
 - It is used to run tests against multiple browsers, multiple versions of browser, and browsers running on different operating systems.

- It reduces the time it takes for the test suite to complete a test pass. A grid consists of a single hub, and one or more nodes.
- The hub receives a test to be executed along with information on which browser and 'platform' where the test should be run.
- It 'knows' the configuration of each node that has been 'registered' to the hub.
- Using this information it selects an available node that has the requested browser-platform combination.
- Once a node has been selected, Selenium commands initiated by the test are send to the hub, which passes them to the node assigned to that test.
- The node runs the browser, and executes the Selenium commands within that browser against the application under test

Q.14 Explain difference between Selenium IDE, Selenium RC and Selenium webDriver.

Ans. :

Selenium IDE	Selenium RC	Selenium webDriver
Runs only on Firefox	Support all type of browsers	Support all type of browsers
Used only as prototyping tool	Used to write advanced scripts	Used to write advanced scripts
It is independent of Selenium server to run the test script.	It uses the Selenium server before processing the test script.	Selenium server isn't needed anymore to run the test script
It brings one of the most valuable record and playback feature.	You can't use it for recording and playback	You can't use it for recording and playback
It does not support listeners.	It does not support listeners	It supports listeners

Core engine is JavaScript	Core engine is JavaScript based	Interacts natively with browser.
It is not object oriented	Its API is slightly object oriented and redundant	Its API is purely object oriented and compact.
It has a UI interface to work with the test scripts	It's a standalone Java application which runs Html test suites in the browser	It is a full-fledged API and languages like Java, Python, and CSharp implement Webdriver APIs.
Very easy to use and install	Complicated configuration	Less complicated configuration than RC

4.6 : Test Design Considerations

Q.15 Explain types of tests. How static content are tested ?

Ans. : Types of Tests :

- What parts of application should be tested ? It depends on aspects of the project like user expectations, time allowed for the project, priorities set by the project manager etc.
- Once the project boundaries are defined though, the tester, will certainly make many decisions on what to test.

Testing Static Content :

- The simplest type of test, a content test, is a simple test for the existence of a static, non-changing, UI element. Following parameters are considered :
 1. Does each page have its expected page title ? This can be used to verify your test found an expected page after following a link.
 2. Does the application's home page contain an image expected to be at the top of the page ?

3. Does each page of the website contain a footer area with links to the company contact page, privacy policy, and trademarks information ?
 4. Does each page begin with heading text using the <h1> tag? And, does each page have the correct text within that header?
- Tester may or may not need content tests. If page content is not likely to be affected then it may be more efficient to test page content manually.
 - If, for example, application involves files being moved to different locations, content tests may prove valuable.

END... ↵

5

UNIT - V

Quality Management

5.1 : Software Quality Concept

Q.1 Define software quality ? Explain quality as a pragmatic view.

Ans. : • Software quality can be defined as : An effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it.

- An effective software process establishes the infrastructure that supports any effort at building a high quality software product.
- A useful product delivers the content, functions, and features that the end-user desires, but as important, it delivers these assets in a reliable, error free way.
- By adding value for both the producer and user of a software product, high quality software provides benefits for the software organization and the end-user community.
- An effective software process establishes the infrastructure that supports any effort at building a high quality software product.
- The management aspects of process create the checks and balances that help avoid project chaos, a key contributor to poor quality.
- Software engineering practices allow the developer to analyze the problem and design a solid solution, both critical to building high quality software.

(5 - 1)

• Finally, umbrella activities such as change management and technical reviews have as much to do with quality as any other part of software engineering practice.

Quality as a Pragmatic View

1. The transcendental view argues that quality is something that you immediately recognize, but cannot explicitly define.
2. User view sees quality in terms of an end-user's specific goals. If a product meets those goals, it exhibits quality.
3. The manufacturer's view defines quality in terms of the original specification of the product. If the product conforms to the spec, it exhibits quality.
4. The product view suggests that quality can be tied to inherent characteristics (e.g., functions and features) of a product.
5. Finally, the value-based view measures quality based on how much a customer is willing to pay for a product. In reality, quality encompasses all of these views and more.

Q.2 Why do you required software quality ? Explain cause of software errors.

Ans. : • Software is computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

- Software quality, by definition, is the degree to which software possesses a desired combination of attributes.
- Software quality is the degree to which a system, component, or process meets specified requirements.
- Software quality is the degree to which a system, component, or process meets customer or user needs or expectations. Quality means conformance to requirements
- Quality that is defined as a matter of products and services whose measurable characteristics satisfy a fixed specification - that is, conformance to an in beforehand defined specification.

- The nine causes of software errors are :
 - Faulty requirements definition
 - Client-developer communication failures
 - Deliberate deviations from software requirements
 - Logical design errors
 - Coding errors
 - Non-compliance with documentation and coding instructions
 - Shortcomings of the testing process
 - User interface and procedure errors
 - Documentation errors.
- Meeting customer needs : Quality that is identified independent of any measurable characteristics. That is, quality is defined as the products or services capability to meet customer expectations explicit or not.
- The fundamental reason for measuring software and the software process is to obtain data that helps us to better control the schedule, cost, and quality of software products.
- It is important to be able to consistently count and measure basic entities that are directly measurable, such as size, defects, effort, and time

Q.3 How organizations achieve software quality ?

- Ans. :
- Software quality is the result of good project management and solid engineering practice. To build high quality software you must understand the problem to be solved and be capable of creating a quality design that conforms to the problem requirements
 - Eliminating architectural flaws during design can improve quality.
 - Project management – Project plan includes explicit techniques for quality and change management
 - Quality control - Series of inspections, reviews, and tests used to ensure conformance of a work product to its specifications.
 - Quality assurance - Consists of the auditing and reporting procedures used to provide management with data needed to make proactive decisions.

Impact of Management Actions

- Estimation decisions : Irrational delivery date estimates cause teams to take short-cuts that can lead to reduced product quality.
- Scheduling decisions : Failing to pay attention to task dependencies when creating the project schedule may force the project team to test modules without their subcomponents and quality may suffer.
- Risk-oriented decisions : Reacting to each crisis as it arises rather than building in mechanisms to monitor risks and having established contingency plans may result in products having reduced quality.

5.2 : Software Quality Assurance

Q.4 Explain difference between SQA and SQC.

Ans. :

Software quality assurance	Software quality control
SQA is pro-active means it identifies weaknesses in the processes.	SQC is reactive means it identifies the defects and also corrects the defects or bugs also.
A set of activities designed to evaluate the process by which the products are developed or manufactured.	Quality control is defined as "a set of activities designed to evaluate the quality of a developed or manufactured product".
Quality assurance is the process of managing for quality.	Quality control is used to verify the quality of the output.
It does not involve executing the program or code.	It always involves executing the program or code.
Verification is an example of quality assurance.	Validation/Software testing is an example of quality control.
Quality assurance is process oriented.	Quality control is product oriented.

In order to meet the customer requirements QA defines standards and methodologies.	QC confirms that the standards are followed while working on the product.
It requires involvement of the whole team.	It requires involvement of testing team.

Q.5 Describe elements of SQA.

Ans. : • Software quality assurance elements are as follows :

- a. Standards
 - b. Reviews and Audits
 - c. Testing
 - d. Error/defect collection and analysis
 - e. Change management
 - f. Education
 - g. Vendor management
 - h. Security management
 - i. Safety
 - j. Risk management
1. Standards : It ensures that standards are adopted and followed.
 2. Reviews and audits : Audits are reviews performed by SQA personnel to ensure that quality guidelines are followed for all software engineering work.
 3. Testing ensure that testing id properly planned and conducted.
 4. Error/defect collection and analysis: it collects and analyses error and defect data to better understand how errors are introduced and can be eliminated.
 5. Changes management ensures that adequate change management practices have been instituted.
 6. Education takes lead in software process improvement and educational program.
 7. Vendor management suggests specific quality practices vendor should follow and incorporates quality mandates in vendor contracts
 8. Security management ensures use of appropriate process and technology to achieve desired security level.
 9. Safety is responsible for assessing impact of software failure and initiating steps to reduce risk.

10. Risk management ensures risk management activities are properly conducted and that contingency plans have been established.

Q.6 Elaborates SQA tasks.

Ans. : SQA Tasks are as follows :

1. Prepare SQA plan for the project. The plan identifies an evaluations to be performed, audits and reviews to be performed, standards that are applicable to the project, procedures for error reporting and tracking, documents to be produced by the SQA group and amount of feedback provided to the software project team.
2. Participate in the development of the project's software process description. The SQA group reviews the process description for compliance with organizational policy, internal software standards, externally imposed standards and other parts of the software project plan.
3. Review software engineering activities to verify compliance with the defined software process.
4. Audit designated software work products to verify compliance with those defined as part of the software process.
5. Ensure that any deviations in software or work products are documented and handled according to a documented procedure.
6. Record any evidence of noncompliance and reports them to management.

5.3 : Statistical Software Quality Assurance**Q.7 Write short note on statistical software quality assurance.**

Ans. : • Information about software errors and defects is collected and categorized. An attempt is made to trace each error and defect to its underlying cause, for example: non-conformance to specifications, design error, violation of standards, poor communication with the customer.

- Using the Pareto principle (80 percent of the defects can be traced to 20 percent of all possible causes), isolate the 20 percent (the vital few).
- Once the vital few causes have been identified, move to correct the problems that have caused the errors and defects

Software Reliability

- It is defined as the probability of failure free operation of a computer program in a specified environment for a specified time period
- It can be measured directly and estimated using historical and developmental data. Software reliability problems can usually be traced back to errors in design or implementation.
- Reliability metrics are units of measure for system reliability.
- System reliability is measured by counting the number of operational failures and relating these to demands made on the system at the time of failure
- A long-term measurement program is required to assess the reliability of critical systems

5.3 : Six Sigma for Software Engineering

Q.8 What is six sigma ? Explain features of six sigma.

- Ans. :
- According to David Card, Six sigma is "A generic quantitative approach to improvement that applies to any process."
 - Six Sigma is a disciplined, data-driven approach and methodology for eliminating defects in any process from manufacturing to transactional and from product to service.
 - In essence, six sigma is an approach to finding the cause of business problems and solving them making an impact on the bottom line of a corporation and decreasing variation for products.

- Since the early 1990, Six Sigma has been extensively used for achieving total customer satisfaction with innovative products at competitive price.
- Its objectives are to deliver products when promised, without delivered defects, early life failures, or failures during use.
- Key Definitions :
 1. Critical to Quality : Internal critical quality parameters that relate to the wants and needs of the customer.
 2. Critical to Customer : The input to the quality function deployment activity.
 3. Defect : Any type of undesired result.
 4. Opportunity : area within a product, process, service, or other system where a defect could be produced or where you fail to achieve the ideal product in the eyes of the customer.

Features of Six Sigma :

1. Six Sigma's aim is to eliminate waste and inefficiency, thereby increasing customer satisfaction by delivering what the customer is expecting.
2. Six Sigma follows a structured methodology, and has defined roles for the participants.
3. Six Sigma is a data driven methodology, and requires accurate data collection for the processes being analyzed.
4. Six Sigma is about putting results on financial statements.
5. Six Sigma is a business-driven, multi-dimensional structured approach.

Q.9 Why do we need Six Sigma For Software ?

- Ans. :
- Software today is responsible for most of the added value in products, and must be blamed for many of its failures. Even if the iron hook breaks, it may be the software embedded in the measurement instrument to blame for not having detected it in time.

- When in Germany the high speed intercity express train crashed into an overpass, it was software that didn't detect the broken wheel ring well before the accident.
- Mobile networks are suffering from not being able to provide interconnection to the Internet and interoperability between their own services. It is the software that fails.
- In e-Commerce and for making Web Services to work, security, reliability and fault tolerance are of essence. Software and business processes are not cooperating, as they should to make it profitable.
- Software is so ubiquitous that we must solve the software development problem to address a lot of other problems the society has.

Q.10 Explain phases of six sigma approach. What are benefits of six sigma ?

Ans. : • The Six Sigma approach is :

1. Set the goal - Define
 2. Define the metrics - Measure
 3. Measure where you go - Analyse
 4. Improve your processes while you go - Improve
 5. Act immediately if going the wrong path - Control.
- The Six Sigma methodology defines three core steps :
 1. Define customer requirements and deliverables and project goals via well-defined methods of customer communication
 2. Measure the existing process and its output to determine current quality performance (collect defect metrics)
 3. Analyze defect metrics and determine the vital few causes.
 4. Improve the process by eliminating the root causes of defects.
 5. Control the process to ensure that future work does not reintroduce the causes of defects. - Benefits of Six Sigma

Six Sigma offers six major benefits that attract companies ?

1. Generates sustained success
2. Sets a performance goal for everyone
3. Enhances value to customers
4. Accelerates the rate of improvement
5. Promotes learning and cross-pollination
6. Executes strategic change.

5.5 : ISO 9000 Quality Standards

Q.11 What is role of ISO 9000 towards SQA standardization ?

Ans. : ISO 9000

- ISO 9000 is an international set of standards for quality management. It is applicable to a range of organizations from manufacturing to service industries.
- ISO 9000 has a broad scope : Hardware, software, processed materials and services. The standard outlines the basic elements of a good quality management system. These elements are good business practice.
- ISO 9001 is applicable to organizations which design, develop and maintain products. ISO 9001 is a generic model of the quality process must be instantiated for each organization.
- ISO 9000-1 provides guidelines for selection and use of the ISO 9000 standards.
- The ISO 9004 series provide guidance for quality management, i.e. for the design, implementation and improvement of a quality system. A quality system is a set of 'organizational structures, procedures, processes and resources needed to implement quality management.
- ISO 9001, 9002 and 9003 are models for external quality assurance. They specify a set of requirements.

- ISO 9000 and ISO 9004 are guidelines for quality management and are not Mandatory for certification.
- ISO 9001, ISO 9002 and ISO 9003 are quality system standards.
- ISO 9001 is the broadest standard and provides a model for design, development, production, installation and servicing.
- ISO 9002 is limited to production, installation and servicing.
- ISO 9003 is further limited to inspection and testing.
- A company should first use ISO 9000 to design and to implement a quality system. Once the quality has been installed, the company may use the quality assurance models of ISO 9001, ISO 9002 or ISO 9003 to demonstrate the adequacy of the quality system.
- ISO 9002 is used for production and installation. It is the standard that governs the manufacture of a product. It is designed to ensure conformance to production and installation methods.
- ISO 9003 is the standard directed at the final test and inspection of products. The standard pre-assumes an extensive quality control function and specifies what is needed for conformance to requirements.
- ISO 9004 is for internal use only and lists the components that compose quality systems.

Why ISO 9000 certification ?

1. Better organizational definition
2. Greater quality awareness
3. Better documentation of processes
4. Increased control of operations
5. Ongoing analysis of and solution to problems
6. Positive cultural change
7. Improved customer satisfaction and increased market opportunities.

ISO 9000 certification

- Quality standards and procedures should be documented in an organisational quality manual.
- External body may certify that an organisation's quality manual conforms to (ISO 9000) standards.
- Customers are, increasingly, demanding that suppliers are ISO 9000 certified.

Q.12 Briefly explain ISO 9000-3 quality management system principle.

Ans. :

1. Customer focus
2. Leadership
3. Involvement of people
4. Process approach
5. System approach to management
6. Continual improvement
7. Factual approach to decision making
8. Mutually supportive supplier relationships

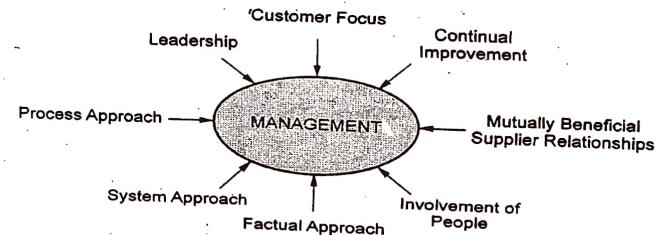


Fig. Q.12.1 The eight quality management principles

- The eight principles of quality management can be used by senior management as a framework to guide their organizations towards improved performance.

Principle	Functions	Benefit
Customer focus	<ul style="list-style-type: none"> Organizations depend on their customers and therefore should understand current and future customer needs. Meet customer requirements and strive to exceed customer expectations. 	<ol style="list-style-type: none"> Increased revenue and market share. Increased effectiveness in the use of the organization's resources. Improved customer loyalty leading to repeat business.
Leadership	<ul style="list-style-type: none"> Leaders establish unity of purpose and direction of the organization. They should create and maintain the internal environment in which people can become fully involved in achieving the organization's objectives. 	<ol style="list-style-type: none"> People will understand and be motivated towards the organization's goals and objectives. Mis-communication will be minimized.
Involvement of people	<ul style="list-style-type: none"> People at all levels are the essence of an organization and their full involvement enables their abilities to be used for the organization's benefit. 	<ol style="list-style-type: none"> Motivated, committed and involved people within the organization.
Process approach	<ul style="list-style-type: none"> A desired result is achieved more efficiently when activities and related resources are managed as a process. 	<ol style="list-style-type: none"> Lower costs and shorter cycle times through effective use of resources. Improved, consistent and predictable results

System approach to management	<ul style="list-style-type: none"> Identifying, understanding and managing interrelated processes as a system contributes to the organization's effectiveness and efficiency in achieving its objectives. 	<ol style="list-style-type: none"> Ability to focus effort on the key processes. Providing confidence to interested parties as to the consistency, effectiveness and efficiency of the organization.
Continual improvement	<ul style="list-style-type: none"> Continual improvement of the organization's overall performance should be a permanent objective of the organization. 	<ol style="list-style-type: none"> Flexibility to react quickly to opportunities. Performance advantage through improved organizational capabilities.
Factual approach to decision making	<ul style="list-style-type: none"> Effective decisions are based on the analysis of data and information. 	Increased ability to review, challenge and change opinions and decisions.
Mutually supportive supplier relationships	<ul style="list-style-type: none"> An organization and its suppliers are interdependent and a mutually beneficial relationship enhances the ability of both to create value. 	<ol style="list-style-type: none"> Increased ability to create value for both parties. Optimization of costs and resources.

Q.13 Explain ISO 9000-3 requirement process.

Ans. : The ISO 9000-3 includes about 20 requirements that relate to various aspects of software quality management classified into the following five groups :

1. Quality management system
2. Management responsibilities
3. Resource management

4. Product realization

5. Measurement, analysis and improvement

Requirements	Functions
Quality management system requirements	<ol style="list-style-type: none"> Establish a quality management system Document the quality management system
Management responsibilities	<ol style="list-style-type: none"> Management commitment Customer focus Quality policy Planning Administration Management review
Resource management	<ol style="list-style-type: none"> Provision of resources Human resources Facilities Work environment
Product realization	<ol style="list-style-type: none"> Planning of realization processes Customer-related processes Design and/or development Purchasing Production and service operations Control of measuring and monitoring devices
Measurement, analysis and improvement	<ol style="list-style-type: none"> Planning Measurement and monitoring Control of nonconformity Analysis of data Improvement

Q.14 Explain certification process of ISO 9000-3.

Ans. : • The ISO 9000-3 certification process verifies that an organization's software development and maintenance processes fully comply with the standard's requirements.

- The objective of the International Standards Organization 9000 series of standards is to certify that an organization has quality manufacturing processes. Thus, if a supplier wanted to demonstrate competence of products, it would make application to demonstrate that it meets the ISO 9000 standards.

- Documentation requirements :** The ISO 9000 audit is heavily focused on evaluation of documentation. Four tiers of quality system documentation are required :

- First tier-the quality manual.
- Second tier-quality management procedures (i.e., core procedures).
- Third tier-area work instructions (i.e., standard operating procedures, test methods, calibration methods).
- Fourth tier-forms, records, books and files.

Certification process :

- The company first implements the control and documentation procedures outlined in the series.
- It then involves a thorough audit by an independent certification organization that is licensed to register quality systems by an accreditation body.
- Upon compliance, it receives a registration certificate and its name is included in a published directory of registered suppliers.
- The systems will be continually verified by the registrar in periodic surveillance and full audits are conducted every few years.

Development of the organization's SQA system :

- Organization first development of a quality model and SQA procedures. Second steps is to development of other SQA infrastructure like staff training, preventive and corrective actions procedures, configuration management services and documentation and quality record control.

- Lastly the development of a project progress control system.
- Implementation of the organization's SQA system :**
- Setting up a staff instruction program
 - Leaders and managers are expected to follow up and support the implementation efforts made by their units.
 - Internal quality audits are carried out to verify the success in implementation.
 - The findings will determine of whether the organization has reached a satisfactory level of implementation.

Problems with certification

1. Costs - application and maintenance
2. Time - application and maintenance
3. Level of internal expertise
4. Executive commitment
5. Selection of registration.

END... ↵

UNIT - VI

6

Software Quality Tools

6.1 : Total Quality Management

Q.1 Explain component involves in TQM.

Ans. : • TQM involves following basic concept :

- | | |
|-------------------------|-----------------------------------|
| 1. Leadership | 2. Customer Satisfaction |
| 3. Employee Involvement | 4. Continuous Process Improvement |
| 5. Supplier Partnership | 6. Performance Measures. |

• **Leadership** : Top management must realize importance of quality. Quality is responsibility of everybody, but ultimate responsibility is CEO. Quality excellence must become part of business strategy. Lead in the implementation process.

• **Customer Satisfaction** : Customer is always right, in Japan customer is "King". Customer expectations constantly changing.

• Satisfaction is a function of total experience with organization. It must give customers a quality product or service, reasonable price, on-time delivery, and outstanding service.

• It need to continually examine the quality systems and practices to be responsive to ever changing needs, requirements and expectations.

• **Employee Involvement** : People are the most important resource/asset. Quality comes from people.

• **Continuous Process Improvement** : View all work as process, production and business. Process means purchasing, design, invoicing.

- Supplier Partnership :** The 40 % product cost comes from purchased materials; therefore supplier quality management is important.
- The substantial portion of quality problems comes from suppliers and need partnership to achieve quality improvement, i.e. long-term purchase contract. It is supplier management activities.
- Performance Measures :** It is managing by fact rather than feelings. The effective management requires measuring performance. It uses a baseline, to identify potential projects, to assess results from improvement.

6.2 : Product Quality Metrics

Q.2 Explain help desk quality metric. Also explain benefit of product metrics.

Ans. : • Product metrics refer to the software system's operational phase. In most of the project, software developer is required to provide customer service during the software's operational phase.

- Customer services are divided into two categories:

- Help Desk services (HD)
- Corrective maintenance services

- HD metrics are based on all customer calls while corrective maintenance metrics are based on failure reports.

- HD quality metrics are as follows:

- HD calls density metrics - Measured by the number of calls.
- HD calls severity metrics - The severity of the HD issues raised.
- HD success metrics - The level of success in responding to HD calls

HD calls density metrics :

$$\text{HD calls Density (HDD)} = \frac{\text{NHYC}}{\text{KLMC}}$$

$$\text{WHDD Weighted HD calls Density WHDD} = \frac{\text{WHYC}}{\text{KLMC}}$$

$$\text{WHDF Weighted HD calls per Function point WHDF} = \frac{\text{WHYC}}{\text{NMFP}}$$

- NHYC = Number of HD calls during a year of service.
- KLMC = Thousands of lines of maintained software code.
- WHYC = Weighted HD calls received during one year of service.
- NMFP = Number of function points to be maintained.

Severity of HD Calls Metrics and HD Success Metrics

Average severity of HD calls (ASHC)

$$= \frac{\text{Weighted HD calls received during one year of service (WHYC)}}{\text{Number of HD calls during a year of service (NHYC)}}$$

HD service success (HDS)

$$= \frac{\text{Number of yearly HD calls completed on time during one year of service (NHYNOT)}}{\text{Number of HD calls during a year of service (NHYC)}}$$

HD productivity and effectiveness metrics

- Productivity metrics relate to the total of resources invested during a specified period. It uses the easy-to-apply KLMC measure of maintained software system's size.
- Effectiveness metrics relate to the resources invested in responding to a HD customer call.

$$\text{HD Productivity (HDP)} = \frac{\text{Total yearly working hours invested in HD servicing of the software system (HDYH)}}{\text{Thousands of lines of maintained software code (KLMC)}}$$

Failures of maintenance services metrics

- A customer call related to a software failure problem that was supposed to be solved after a previous call is commonly treated as a maintenance service failure.

Maintenance repeated repair failure (MRepF)

$$\text{Number of repeated software failure calls (RepYF)} = \frac{\text{Number of software failures detected during a year of maintenance service (NYF)}}{\text{Number of software failures detected during a year of maintenance service (NYF)}}$$

Benefits of product metrics

1. Assist in the evaluation of the analysis and evaluation model
2. Provide indication of procedural design complexity and source code complexity
3. Facilitate design of more effective testing

6.3 : In Process Quality Metrics**6.3 Explain error density metric and error severity metrics.****Ans. :****Error density metrics**

- Calculation of error density metrics involves two measures : Software volume and errors counted.
- Software volume measures. Some density metrics use the number of lines of code while others apply function points.
- Errors counted measures. Some relate to the number of errors and others to the weighted number of errors.

Name	Formula
Code Error Density	CED = NCE / KLOC
Development Error Density	DED = NDE / KLOC
Weighted Code Error Density	WCED = WCE / KLOC
Weighted Development Error Density	WDDED = WDE / KLOC
Weighted Code Errors per Function Point	WCEF = WCE / NFP
Weighted Development Errors per Function Point	WDEF = WDE / NFP

NCE = The number of code errors detected by code inspections and testing.

NCE = The number of code errors detected by code inspections and testing.

NDE = Total number of development (design and code) errors detected in the development process.

WCE = Weighted total code errors detected by code inspections and testing.

WDE = Total weighted development (design and code) errors detected in development process.

- A software development department may apply two alternative metrics for calculation of code error density : CED and WCED.
- While the CED does not indicate quality below the acceptable level, the WCED metric does indicate quality below the acceptable level.

Error severity metrics

- The metrics belonging to this group are used to detect adverse situations of increasing numbers of severe errors in situations where errors and weighted errors, as measured by error density metrics, are generally decreasing

Code	Formula
Average Severity of Code Errors	ASCE = WCE / NCE
Average Severity of Development Errors	ASDE = WDE / NDE

Q.4 What is error removal effectiveness metrics ?

Ans. : • Software developers can measure the effectiveness of error removal by the software quality assurance system after a period of regular operation of the system.

$$\text{Development Errors Removal Effectiveness (DERE)} = \frac{\text{NDE}}{\text{NDE} + \text{NYF}}$$

$$\text{Development Weighted Errors Removal Effectiveness (DWERE)} = \frac{\text{WDE}}{\text{WDE} + \text{WYF}}$$

- NDE = Total number of development (design and code) errors detected in the development process.
- WCE = Weighted total code errors detected by code inspections and testing.
- WDE = Total weighted development (design and code) errors detected in development process.
- NYF = Number software failures detected during a year of maintenance service.
- WYF = Weighted number of software failures detected during a year of maintenance service.

6.4 : Software Maintenance

Q.5 What is software maintenance ? Explain different types of maintenance.

Ans. : • **Definition of maintenance :** It is the set of activities, both technical and managerial, that ensures that software continues to meet organizational and business objectives in a cost effective way.

Types of maintenance

1. **Corrective :** Taking existing code and correcting a fault that causes the code to behave in some way that deviates from its documented requirements.
2. **Adaptive :** Taking existing code and adapting it to provide new features and functionality. These are typically part of a new release of the code and part of a larger development effort.

3. **Perfective :** These are typically made to improve the maintainability of the code such as restructuring it to make it more easily understood or to remove ambiguities.
4. **Inspection :** These are usually made as a result of code inspections and focus more of adhering to coding standards or to reduce the likelihood of a failure.

Q.6 Briefly explain maintenance contract review.

Ans. : Maintenance contract review :

- When considering the maintenance contract, many issues need to be addressed. These issues and decisions can vary from the type of customer or software the contract review is for.
- These are just two reasons that an adequate maintenance contract needs to be drawn up before agreeing to supply software maintenance services to any type of customer. The main objectives of software maintenance contract reviews are :
 1. **Customer requirements classification :** Issues such as; hours of service, size of user population, location of users and the types of applications to be used, all need to be considered.
 2. **Review of alternative approaches to maintenance provision :** such as subcontracting types of services.
 3. **Review of estimates of required maintenance resources :** Firstly these estimates should be examined with regards to advice from the proposal team about the required maintenance services. After this, the estimates should be reviewed with consideration to the company's ability to meet commitments, with respect to the professional competencies as well as the maintenance teams' abilities.
 4. **Review of maintenance services to be provided by subcontractors and/or the customer :** This review allows each party involved, to assess the services provided by one-another. Issues such as; quality assurance, payment to subcontractors and follow-up procedures are reviewed.

5. Review of maintenance cost estimates : "These estimates should be reviewed on the basis of required resources."

Q.7 What is maintenance plan ? explain.

Ans. : • Maintenance plan is prepared for all projects and customers. The plan provide the framework within which maintenance provision is organized.

- The plan contains the following :

1. A list of the contracted maintenance services : It includes name of the internal and external customers, the number of users, characteristics of corrective maintenance services (remote and on site) etc.
2. A description of the maintenance team's organization.
3. A list of maintenance facilities : This is the infrastructure, which includes the maintenance support centre and a documentation centre that makes it possible to provide a good service.
4. A list of identified maintenance service risks : The list of identified maintenance service risks refers to situations whereby the inability to provide sufficient maintenance is anticipated.
5. A list of required software maintenance procedures and controls.
6. The software maintenance budget

6.5 : Ishikawa's 7 Basic Tools

Q.8 What is a Pareto diagram? Explain.

Ans. : • Pareto diagram is a frequency chart of bars in descending order. This means the categories represented by the tall bars on the left are relatively more significant than those on the right. This bar chart is used to separate the "vital few" from the "trivial many".

- Fig. Q.8.1 shows pareto diagram.
- The aim of Pareto chart is to figure out the different kind of "nonconformity" from data figures, maintenance data, repair data, parts scrap rates, or other sources.

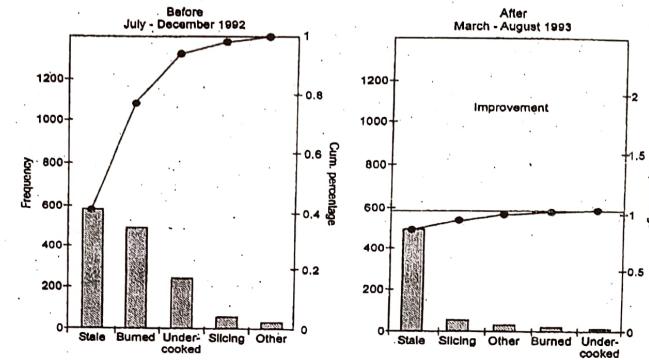


Fig. Q.8.1 Pareto diagram

- Pareto chart can generate a mean for investigating concerning quality improvement and improving efficiency, "material waste, energy conservation, safety issues, cost reductions", etc.
- In software development, the X-axis for a Pareto diagram is usually the defect cause and the Y-axis the defect count. By arranging the causes based on defect frequency, a Pareto diagram can identify the few causes that account for the majority of defects. It indicates which problems should be solved first in eliminating defects and improving the operation.
- Pareto analysis is commonly referred to as the 80 "20 principle (20 % of the causes account for 80 % of the defects), although the cause-defect relationship is not always in an 80 "20 distribution.

Q.9 What is a cause-effect diagram ? Explain.

Ans. : • The cause-and-effect diagram, also known as the fishbone diagram, was developed by Ishikawa and associates in the early 1950s in Japan.

- It has also two other names that are Ishikawa diagram and fishbone because the shape of the diagram looks like the skeleton of a fish to identify quality problems based on their degree of importance.

- Fig. Q.9.1 shows cause effect diagram.

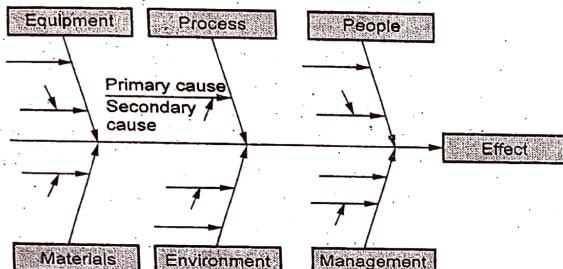


Fig. Q.9.1 Cause effect diagram

- The cause and effect diagram is used to explore all the potential or real causes (or inputs) that result in a single effect (or output).
- Causes are arranged according to their level of importance or detail, resulting in a depiction of relationships and hierarchy of events. This can help you search for root causes, identify areas where there may be problems and compare the relative importance of different causes.

Q.10 Write short note on a) Scatter diagram b) Checklist c) Run chart

Ans. :

a) Scatter diagram :

Scatter diagram is a powerful tool to draw the distribution of information in two dimensions. It helps to detect and analyze pattern relationships between two quality and compliance variables. It understanding the relationship between them, like Weak or strong and positive or negative.

- Fig. Q.10.1 shows scatter diagram.

- The shape of the scatter diagram often shows the degree and direction of relationship between two variables, and the correlation may reveal the causes of a problem.

- In these diagrams, one variable denotes one axis and another variable denotes the other axis.
- Correlation refers to the measure of the relationship between two sets of numbers or variables.
- However, correlation does not necessarily mean a direct cause and effect relationship. If it appears that values for one of the variables can be predicted based on the value of the other variable, then there is correlation.

b) Checklist :

- A check sheet can be introduced as the most basic tool for quality. It is basically used for gathering and organizing data. Fig. Q.10.2 shows check sheet.

Reason	Day					
	Mon	Tues	Wed	Thus	Fri	Total
Wrong number	III	II	I	III	III	20
Info request	II	II	II	II	II	10
Boss	III	II	III	I	III	19
Total	12	6	10	8	13	49

Fig. Q.10.2 Checklist/checksheet

- Data are "collected and tabulated" on the check sheet to record the frequency of specific events during a data collection period. They prepare a "consistent, effective and economical approach".

that can be applied in the auditing of quality assurance for re-viewing and to follow the steps in a particular process

- When this is done with the help of software packages such as Microsoft Excel, you can derive further analysis graphs and automate through macros available.
- The check sheets are in several, three major types are such as Defect-location check sheets; tally check sheets, and; defect-cause check sheets.

Advantages

- The main advantages of check sheets are to be very easily to apply and understand.
- It can make a clear picture of the situation and condition of the organization.
- They are efficient and powerful tools to identify frequently problems.

Disadvantages

- They do not have effective ability to analyze the quality problem into the workplace.

c) Run chart :

- A run chart tracks the performance of the parameter of interest over time. The X-axis is time and the Y-axis is the value of the parameter.
- A run chart is best used for trend analysis, especially if historical data are available for comparisons with the current trend.
- Ishikawa includes various graphs such as the pie chart, bar graph, compound bar graph, and circle graph under the section that discusses run charts .
- An example of a run chart in software is the weekly number of open problems in the backlog; it shows the development team's workload of software fixes.
- Fig. Q.10.3 shows run time chart.

- An organization's desire is to have their product arrive to their

Run chart/Time plots

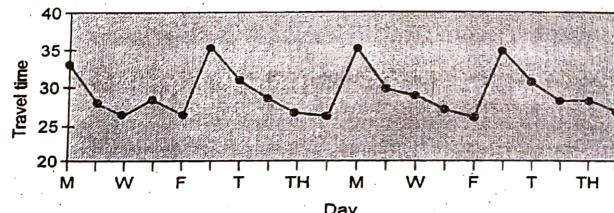


Fig. Q.10.3 Run chart

customers on time, but they have noticed that it doesn't take the same amount of time each day of the week.

- They decided to monitor the amount of time it takes to deliver their product over the next few weeks.

6.6 : Defect Removal Effectiveness

Q.11 Describe simple defect tracking process steps.

Ans. : The steps below describe a simple defect tracking process :

- Execute the test and compare the actual results to the documented expected results.
- If a discrepancy exists, log the discrepancy with a status of "open". Supplementary documentation, such as screen prints or program traces, should be attached if available.
- The test manager or tester should review the problem log with the appropriate member of the development team to determine if the discrepancy is truly a defect.
- Assign the defect to a developer for correction.

5. Once the defect is corrected, the developer will usually enter a description of the fix applied and update the defect status to "Fixed" or "Retest".
6. The defect is routed back to the test team for retesting.
7. Additional regression testing is performed as needed based on the severity and impact of the fix applied.
8. If the retest result match the expected result, the defect status is updated to "closed". If the test results indicate that the defect is still not fixed, the status is changed to "open" and sent back to the developer.

Q.12 List and explain classification of defects.

Ans. : • Defects are of different types. Some defects can lead to method failure. Some defects can be mild. Thus, the defects must be classified according to its impact on the functionality of the application. Defects can be classified as following are of different types of defects :

1. **Critical** : The defects termed as 'critical', needs immediate attention and treatment. A critical defect directly affects the critical and essential functionalities, which may affect a software product or its functionality on a large scale, such as failure of a feature/functionality or the whole system, system crash-down, etc.
2. **Major** : Defects, which are responsible for affecting the core and major functionalities of a software product. Although, these defects does not results into complete failure of a system, but may bring several major functions of software, to rest.
3. **Minor** : These defects produce minor impact, and does not have any significant influence on a software product. The results of these defects may be seen in the product's working, however, it does not stops users to execute it task, which may be carried out, using some other alternative.

4. **Trivial** : These types of defects, have no impact on the working of a product, and sometimes, it is ignored and skipped, such as spelling or grammatical mistake.

Q.13 What is defect removal efficiency ?

Ans. : • Defect removal effectiveness is a very important aspect of product quality. A good defect removal process promotes the release of products with lower latent defects, generating high customer confidence.

- Defect Removal Efficiency (DRE) is a powerful metric used to measure test effectiveness. From this metric we come to know how many bugs we found from the set of bugs which we could have found.
- The following is the formula for calculating DRE.

$$\text{Defect removal efficiency} = \frac{\text{Total defect detected during inspection and testing}}{\text{Total defects detected}} \times 100$$

- Using the above defect removal efficiency metric, reviewers/testers might uncover defects during software development that are not critical but still achieve higher defect removal efficiency.
- In spite of high defect removal efficiency the customer might be unhappy, due to the presence of critical defects in the software.
- A defect found early, or before a customer finds it, has value. To the developer, the value lies in a higher quality product that will have a better reputation, or, when the defect is found early the cost to fix it is substantially less than if found later in the project.
- To the customer, the value of a defect found by the developer is that the customer will not encounter the problems caused by the defect.
- If processes, people, and tools were perfect, perhaps we could achieve "zero defects." However, defects are a fact of life with the resources we work with at present.

- Another given is that defects can cause serious problems in a project if they are not managed properly. Defect correction consumes resources and can even cause new defects.

6.7 : Process Maturity Level

Q.14 Write short note on process maturity level.

- Ans. :
- When project size and complexity is increases, automated software tools are required for controlling progress report.
 - Project management is a challenging task with many complex responsibilities. Fortunately, there are many tools available to assist with accomplishing the tasks and executing the responsibilities. Some require a computer with supporting software, while others can be used manually. Project managers should choose a project management tool that best suits their management style.
 - No one tool addresses all project management needs. Program Evaluation Review Technique (PERT) and Gantt Charts are two of the most commonly used project management tools and are described below. Both of these project management tools can be produced manually or with commercially available project management software.
 - PERT is a planning and control tool used for defining and controlling the tasks necessary to complete a project. PERT charts and Critical Path Method (CPM) charts are often used interchangeably; the only difference is how task times are computed. Both charts display the total project with all scheduled tasks shown in sequence.
 - The displayed tasks show which ones are in parallel, those tasks that can be performed at the same time. A graphic representation called a "Project Network" or "CPM Diagram" is used to portray graphically the interrelationships of the elements of a project and to show the order in which the activities must be performed.
 - Examples of services that computerized tools

1. Computerized control of risk management activities are as follows :
 - a. Category-wise list of software risk and their planned solution dates.
 - b. Lists of exceptions of software risk items
2. Project schedule control
 - a. Delayed activities list
 - b. Lists of delayed milestones
 - c. Lists of delays of critical activities
 - d. Updated activity schedules generated according to progress reports and correction measures applied
3. Computerized project resource control
 - a. Project resources allocation plan
 - For activities and software modules
 - For teams and development units
 - For designated time periods, etc.
 - b. Project resources utilization
 - c. Project resources utilization exceptions
 - d. Updated resource allocation plans generated according to progress reports and reaction measures applied
4. Computerized project budget control
 - a. Project budget plans
 - For activity and software module
 - For teams and development units
 - For designated time periods, etc.
 - b. Project budget utilization reports
 - c. Project budget utilization deviations
 - d. Updated budget plans generated according to progress reports ports and correction measures

END... ↗

Course 2015**Time : 1 Hour]****[Maximum Marks : 30****Instructions to the candidates :**

- i) Answer Q.1 or Q.2, Q.3 or Q4, Q.5 or Q.6.
- ii) Figures to the right indicate full marks.
- iii) Assume suitable data, if necessary.

Q.1 a) Identify the software quality attributes for the following scenarios :

- i) Any popular websites such as Google.com, Amazon.com and YouTube.com is visited by thousands of users concurrently. What could be the top 2 architectural drivers (quality attributes) for this system ? Justify your answer.
- ii) While updating anti-virus software, it take lot of time so the updating tool displays a progress bar showing the percentages of completion of work and how many files are scanned so far. Which quality attribute is being addressed by this tactic ? Justify your answer.
- iii) A software makes use of adaptors to connect to SMS gateways of different service providers such as airtel, vodafone, BSNL, etc. All these adaptors implement the same interface. Depending on the service provider defined in the configuration file, the software instantiates(creates) an object of the appropriate adaptor class at run time and makes use of it to communicate with the SMS gateway.

[6]

Ans. : i) Performance is about timing. Events (interrupts, messages, requests from users, or the passage of time) occur, and the system must respond to them. A performance scenario begins with a request for some service arriving at the system. Satisfying the request requires resources to be consumed. While this is happening the system may be simultaneously servicing other requests.

ii) Modifiability is about the cost of change

iii) Usability is concerned with how easy it is for the user to accomplish a desired task and the kind of user support the system provides. Learning system features. If the user is unfamiliar with a particular system or a particular aspect of it, what can the system do to make the task of learning easier ?

b) Define quality as viewed by different stakeholders of software development and usage. (Refer Q.4 of Chapter - 1) [4]

OR

Q.2 a) Identify the software quality attributes for the following scenarios :

i) Now a days most of the peoples are using Internet banking for online transaction. What could be the top 2 architectural drivers(quality attributes) for this system ? Justify your answer.

ii) Company want build a game for the children's which they should play from any device. Also various input devices (i.e. mouse, joystick, touch screen etc...) may also integrated for the playing a game.

iii) A software company is in a process of building social networking site which will have very large number of users in near future. Also company wish to add new features in this site and during addition of new features site should provide all the current fratures without any disturbane. What top 2 quality attribute is being addressed by this tactic? Justify your answer. [6]

Ans. : i) Security has two major aspects : Confidentiality (access only granted to authorized users) and authenticity (trust the provided information). Reliability reflects the ability of a service to operate correctly.

ii) Testability matters when it comes to building and automating tests of individual components, interactions between components, as well as the system as a whole

iii) Extensibility represents the ability to add functionality to a component without touching other components or parts of the system. Reliability reflects the ability of a service to operate correctly.

b) Give an example quality in different areas of software development, and benchmarking and metrics for the same. [4]

Ans. : • Benchmarking is the process of comparing the cost, cycle time, productivity, or quality of a specific process or method to another that is widely considered to be an industry standard or best practice.

- Essentially, benchmarking provides a snapshot of the performance of your business and helps you understand where you are in relation to a particular standard.
- Benchmarking is most used to measure performance using a specific indicator (cost per unit of measure, productivity per unit of measure, cycle time of x per unit of measure or defects per unit of measure) resulting in a metric of performance that is then compared to others.
- There are four primary types of benchmarking: internal, competitive, functional, and generic.
 1. Internal benchmarking is a comparison of a business process to a similar process inside the organization.
 2. Competitive benchmarking is a direct competitor-to-competitor comparison of a product, service, process, or method.
 3. Functional benchmarking is a comparison to similar or identical practices within the same or similar functions outside the immediate industry.
 4. Generic benchmarking broadly conceptualizes unrelated business processes or functions that can be practiced in the same or similar ways regardless of the industry.

Q.3 Explain following for testing process.

- a) Misconceptions
- b) Principles (Refer Q.12 of Chapter - 2)
- c) Salient and Policy of software testing (Refer Q.25 of Chapter 2)
- d) Test strategy
- e) Test planning

[10]

Ans. : a) Misconceptions : Testing is a Cost Center : This is probably the single most common misunderstanding regarding software testing, shared by executives across all sectors. In their minds, testing is and always will be a cost center. As a result, their primary goal is to keep these costs down as much as possible.

- Another major, common misconception in this area is that legacy tools can do the job when it comes to software testing. Most significantly, a lot of businesses have moved away from the waterfall methodology, and yet, they have not embraced agile software testing tools.

d) Test strategy : Planning a strategy is an essential component in the process of attaining a successful result. One needs to take into account the possible number of risks involved, degree of accuracy of test results etc. A test strategy thus helps to uncover the facts like which type of testing best suits our requirements, the sequence in which the tests are to be performed etc. Basically, a test strategy helps to sort our tasks in a wise manner and that results in less wastage of time and efforts.

e) Testing planning : A test plan is a layout of how a project's testing activities are to be carried out. It is about classifying as to who will be assigned which task, in which order the tasks shall begin, the optimum time allocated to each task etc. A test plan is exercised after the test strategy is in place.

OR

Q.4 a) With respect quality management systems explain the following.

- i) Quality management systems structure
- ii) Pillars of quality management system
- iii) Important aspects of quality management

[6]

Ans. : i) Quality management systems structure

- There is no certain standard for the structure of the quality management system, it just has to be documented. The

documenting can be done in any form as long as it is suitable for the organization's use.

- When the organization is starting to implement the quality management system in its operations, there should be a reason why this is done. The reason should come from top management, and it should not be "because the competitors are doing it or because we want to have quality certification".
- The review of current business is done to clarify the purpose of the organization and to understand the current market position. The following items should be considered, when reviewing the business:

- | | |
|---|---------------------------------|
| 1. Customer | 2. Product |
| 3. Stakeholders | 4. Description of the operation |
| 4. Strengths and weaknesses | 5. Market share |
| 6. Competitors and the state of competition | |

ii) Pillars of Quality Management System

- Quality processes, quality procedures, work instruction and methods are defined at an organization level by the functional area experts, and project and functional level by the experts in those area separately.
- Guidelines and standards: guidelines and standards are used by organization project team for achieving quality goals for the products and services delivered to customers.
- Formats and templates : common format and templates are used for tracking a project, function, information within organizations. It act as checklist to maintain consistency across the projects in the organizations.

iii) Important aspects of quality management

- For a TQM program to be successful, five major things must be in place within an organization. They are :
 1. Continuous improvement

2. Customer satisfaction
 3. Managerial involvement
 4. Measurability; the ability to accurately measure and record quality and defects
 5. Organizational support for total quality
- b) What are types of requirements and product ? Also pointout relationship between quality and productivity.
(Refer Q.20 and Q.21 of Chapter - 1) [4]

Q.5 a) What is automated testing in software testing ? How do we decide what to automate in testing and the scope of automation ?
(Refer Q.1, Q.4 and Q.5 of Chapter - 3) [6]

b) What is JMeter testing ? Explain the features of Jmeter ? At least three. (Refer Q.16 of Chapter - 3) [4]

OR

- Q.6 a) What is test automation. (Refer Q.1 of Chapter - 3)
- b) Different terms used in automation.
- c) Skills needed for automation. (Refer Q.3 of Chapter - 3)
- d) What to automate. (Refer Q.5 (b) of same paper)
- e) How to define scope of automation. [10]

- Ans. : b) Terms used in automation : • Acceptance criteria : The exit criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity.
- Acceptance testing : Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.
- Accessibility testing : Testing to determine the ease by which users with disabilities can use a component or system.

- Ad hoc testing : Testing carried out informally; no formal test preparation takes place, no recognized test design technique is used, there are no expectations for results and randomness guides the test execution activity.
- Adaptability : The capability of the software product to be adapted for different specified environments without applying actions or means other than those provided for this purpose for the software considered.
- A programming (XP), treating development as the customer of testing and emphasizing the test-first design paradigm.
- Alpha testing : Simulated or actual operational testing by potential users/customers or an independent test team at the developers' site, but outside the development organization. Alpha testing is often employed as a form of internal acceptance testing.
- Analyzability : The capability of the software product to be diagnosed for deficiencies or causes of failures in the software, or for the parts to be modified to be identified. [ISO 9126] See also maintainability testing.
- Audit : An independent evaluation of software products or processes to ascertain compliance to standards, guidelines, specifications, and/or procedures based on objective criteria, including documents that specify :
 - 1) The form or content of the products to be produced.
 - 2) The process by which the products shall be produced.
 - 3) How compliance to standards or guidelines shall be measured ?

- e) Scope of automation : The extent to which automation is used in the testing process defines the scope of automated testing. Among system tests, automation best facilitates security, configuration, and load testing. Regression testing also works well with automation during development and at the initial time of release.

- The scope of automation testing has a promising future. Automation testing allows you to test the application with the help of any automation testing tools as per your requirements. But it can also be converted to a tedious work if the tester is unaware of the programming languages

DECEMBER-2018 [END-SEM] [5461]-599

Solved Paper

Course 2015

Time : 2½ Hours]

[Maximum Marks : 70

Instructions to the candidates

- Assume suitable data if necessary. Mention your assumptions.
- Right indicates the full marks and bifurcation for sub questions.
- Draw suitable diagrams and tables if necessary.

Q.1 a) What is bug; defect, error and failure give an example of each ? [6 = 3 + 3]

Ans. : 1. Bugs : A bug is nothing but a programming error found in an application before it goes into production. It may cause the program to perform irrationally, come up with incorrect responses and then crash all together. If testers find any mismatch in the application/system in testing phase then they call it as Bug.

2. Defect : Defect is the difference observed between expected behaviour and the actual results which are noted post the event of testing a software. Essentially it marks the deviation of the product's performance from the customer's requirements. It differs from a bug in the fact that a defect is found out after the product goes into production. If a developer finds an issue and corrects it by himself in the development phase then it's called a defect.

- Error : An error is a programming mistake, generated due to incorrect loops or wrong syntax. Thus the birth of an error takes place at the time of coding itself. Its chief causes are an inadequate understanding of the internal structure of code, neglect on the part of memory management, etc.
- Failure : Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as failure

b) Differentiate between software testing tools and techniques.
(Refer Q.15 of Chapter - 1)

[6 = 3 + 3]

OR

Q.2 a) What is impact of defect in different phases of software development ? (Refer Q.30 of Chapter - 1) [6]

b) What is bug tracking, bug fixing and bug verification.

Ans. : • Bug tracking is a software application that keeps track of reported software bugs in software development projects

• Bug fixing : When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."

• Bug verification is done to verify that bugs are properly fixed and to check that the fixes don't yield any new issues. The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."

Q.3 a) What is mutation testing ? Give an example.
(Refer Q.18 of Chapter - 2)

[7 = 3 + 4]

b) Why independent testing team is required in organizations ?

- Ans. :**
- An independent team is needed who will have no feelings or regards toward the program and will act completely independently. They will try to find bugs and errors in the program from all angles.
 - The independent team of testers does not report to the development or project manager. It reports to an independent organization.
 - The team of such testers would consist of product specialists, specialists in technology, security testers, certification testers, and test automation experts.

Acceptance and Solution Testing

- Another option would be to use a team of testers who are not related to the developmental team and are outside of it. They are only responsible of answering to the project manager.
- While this may seem like an option which provides independent testing, there is always the possibility that time restraints lead the project manager to influence the testers, once again, leading to bias.
- So while this solution would provide a dedicated environment free from the constraints present in a developmental environment, namely lack of users and full integration, a level of bias would remain.

OR

Q.4 a) What are the skill set required by software tester ?

(Refer Q.37 of Chapter - 2)

[7 = 3 + 4]

b) With the suitable sample, explain requirement traceability matrix. (Refer Q.5 of Chapter - 2)

Q.5 a) What are the selection criteria of automated testing tool ?

[7 = 3 + 4]

Ans. : For automating any application, the following parameters should be considered :

- Data driven capabilities
- Debugging and logging capabilities

- Platform independence
- Extensibility and customizability
- E-mail notifications
- Version control friendly
- Support unattended test runs.

Types of Frameworks :

Typically, there are four test automation frameworks that are adopted while automating the applications.

- Data Driven Automation Framework
- Keyword Driven Automation Framework
- Modular Automation Framework
- Hybrid Automation Framework.

b) What are the difference in testing automation in agile and waterfall model ?

Ans. :

- The Waterfall approach is more structured and rigid, while the Agile approach is more flexible and malleable

- For starters, in the Waterfall model, software development is completed in a single project, which is subsequently split into different phases. This can be thought of as a singular race. With Agile, the development is considered to be many different projects that create a cohesive whole in the end. These can be thought of as a series of small sprints.

- In Waterfall, the testing phase happens after the building phase. This means clients must be concise and clear on requirements before development begins. Once the project begins, there cannot be any changes made. Other phases of development, like design or testing, have to be complete once in Waterfall as well.

- In Agile, the testing phase happens in conjunction with programming and there is a much more developed approach. This means that changes can be made throughout and

requirements can change after the planning stage if they need to. Phases like prototyping and planning tend to appear multiple times unlike in the Waterfall model.

OR

Q.6 a) Discuss "automation in extreme programming".

(Refer Q.13 of Chapter - 3)

[7 = 3 + 4]

b) What are some of the challenges in automating the testing of GUI portions of an application? How do these compare with the automation of back-end testing ?

Ans. : Challenges :

1. Application window name is dynamically Changing.
2. Controls which don't have proper and unique text property. (E.g. Text and Combo box)
3. Mixed up of Managed and Unmanaged UI controls.
4. Applications developed in multiple languages and multiple OS.
5. Customized controls and Owner draw controls.
6. Applications that are using third party controls like source grid control.
7. Class names are dynamically changing.
8. Control names are dynamically changing.
9. Controls which don't have proper Z order.
10. More visual controls.
11. Support for Win32 Controls.
12. Synchronization issue between tool and Device under Test.

Q.7 a) What is Selenium ? What are the different Selenium components ? (Refer Q.1 and Q.5 of Chapter - 4) [16=6+5+5]

b) What are the limitations of Selenium ?

(Refer Q.7 of Chapter - 4)

c) Why one should select selenium as a test tool ?

Ans. : • Selenium is a portable software-testing framework for web applications. Selenium provides a record/playback tool for authoring tests without the need to learn a test scripting language.

- Selenium is an open source automation testing tool which is used to test web applications.
- Selenium testing tool is also consists of different set of tools in it like Selenium WebDriver, Selenium RC, Selenium IDE which has different features.
- Selenium works in multiple operating systems
- Supports multiple browsers - You can use it in Chrome, IE, Firefox, Opera, Safari
- It also can be used to test mobile web applications - Test runs on Android, Blackberry, Iphone
- Easy Implementation

OR

Q.8 a) Describe selenium Web driver architecture with the help of neat diagram. [16=6+5+5]

Ans. : • Fig. 1 shows Selenium Webdriver Architecture

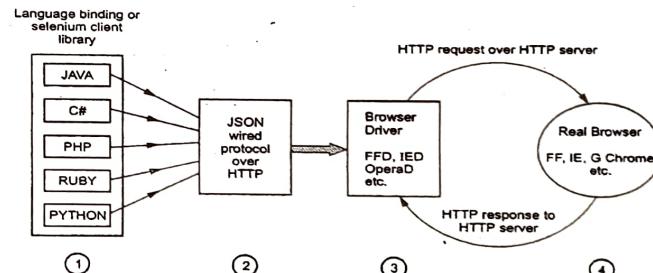


Fig. 1

- selenium Webdriver architecture consists of three layers :
 1. Language binding : To support multiple languages, selenium people has developed language bindings. If you want to use the browser driver in Java, use the Java bindings for Selenium Webdriver. If you want to use the browser driver in C#, Ruby or Python, use the binding for that language. All language binding can be downloaded from selenium official website.
 2. Selenium Webdriver : It is an API which makes possible to communication between programming languages and browsers. It follows object oriented concepts. It has multiple classes and interfaces.
 3. Browser drivers : A browser driver can be considered as a personal secretary of a boss. A browser drivers helps in communication with browser without revealing the internal logic of browser's functionality. The browser driver is the same regardless of the language used for automation.
- When the automation script is executed, the following steps are done internally:
 1. A HTTP request is created and sent to browser driver for each selenium instruction or commands.
 2. A browser driver receives the HTTP request through HTTP server.
 3. HTTP server decides all steps to perform instructions which are executed on browser.
 4. Execution status is sent back to HTTP server which is sent back to automation script.

b) What is Selenium ? What are the different features of it ?

(Refer Q.1 of Chapter - 4)

c) Enlist and explain the components of selenium tool.

(Refer Q.5 of Chapter - 4)

Q.9 a) What does SQA ensure ? What are the goals of SQA ?

[17=6+6+5]

- Ans. : • Software Quality Assurance (SQA) is a process that ensures that developed software meets and complies with defined or standardized quality specifications.
- SQA is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures.

SQA Goals :

- Requirements quality : The correctness, completeness, and consistency of the requirements model will have a strong influence on the quality of all work products that follow.
- Design quality : Every element of the design model should be assessed by the software team to ensure that it exhibits high quality and that the design itself conforms to requirements.
- Code quality : Source code and related work products must conform to local coding standards and exhibit characteristics that will facilitate maintainability.
- Quality control effectiveness : A software team should apply limited resources in a way that has the highest likelihood of achieving a high quality result.

b) State and explain principles of quality management.

(Refer Q.12 of Chapter - 5)

c) What is ISO standard ? What are its advantages ?

Ans. : Refer Q.11 of Chapter - 5.

Advantages :

1. Increased marketability.
2. Reduced operational expenses.
3. Better management control.
4. Improved customer service.
5. Increased customer satisfaction.

OR

Q.10 a) What is six sigma ? Explain the terms DMAIC and DMADV.

[17 = 6+6+5]

- The manufacturer and seller are definitely not the only people who suffer quality-related costs. The customer suffers quality-related costs too. If a manufacturer sells a bad product, the customer faces significant expenses in dealing with that bad product.

Q.11 a) Enumerate Ishikawa's seven basic quality tools. Explain any two in detail. (Refer Q.10 of Chapter - 6) [17=6+6+5]

b) Describe key elements of total quality management.
(Refer Q.1 of Chapter - 6)

c) Explain with example product quality metric.
(Refer Q.2 of Chapter - 6)

OR

Q.12 a) Write short note on Total Quality Management (TQM).

(Refer Q.1 of Chapter - 6 and Q.8 of Chapter - 1) [17=6+6+5]

b) Explain following terms (any two)

Pareto chart (Refer Q.8 of Chapter - 6)

Scatter diagrams (Refer Q.10 of Chapter - 6)

Cause and effect diagrams (Refer Q.9 of Chapter - 6)

c) Describe in detail defect removal effectiveness.

(Refer Q.13 of Chapter - 6)

END... ↗