5-Finally, during execution, tools will run any existing automated tests. These tools follow the defined scripts without deviation. That can seem like an unalloyed “good thing” at first. However, if we did not design the scripts properly, that can mean that the script can get out of sync with the system under test and generate a bunch of false positives.

Q9-what are the activities of the software testing process? explain each in brief

Ans:-1-Testing is a process rather than a single activity.Testing must be planned and it requires discipline to act upon it.The quality and effectiveness of software testing are primarily determined by the quality of the test processes used.
2-The activities of testing can be divided into the following basic steps:

i. Planning and Control
ii. Analysis and Design
iii. Implementation and Execution
iv. Evaluating exit criteria and Reporting
v. Test Closure activities

Planning and Control:-

Test Planning : Test planning involves producing a document that describes an overall approach and test objectives. It involves reviewing the test basis, identifying the test conditions based on analysis of test items, writing test cases and Designing the test environment. Completion or exit criteria must be specified so that we know when testing (at any stage) is complete.

Purpose

- To determine the scope and risks and identify the objectives of testing.
- To determine the required test resources like people, test environments etc.
- To schedule test analysis and design tasks, test implementation, execution and evaluation.

Control:-

This is the activity of comparing actual progress against the plan, and reporting the status, including deviations from the plan. It involves taking actions necessary to meet the mission and objectives of the project.

Analysis and Design

Test analysis and Test Design has the following major tasks:

- To review the test basis. The test basis is the information on which test cases are based, such as requirements, design specifications, product risk analysis, architecture and interfaces
- To identify test conditions
- To design the tests
- To design the test environment set-up and identify the required infrastructure and tools

Implementation and Execution:-

Test execution involves actually running the specified test on a computer system either manually or by using an automated test tool.It is a Fundamental Test Process in which actual work is done.

Test implementation has the following major task:

- To develop and prioritize test cases by using techniques and create test data for those tests.
- To create test suites from the test cases for efficient test execution.
- Test suite is a collection of test cases that are used to test a software program
- To re-execute the tests that previously failed in order to confirm a fix.
- To log the outcome of the test execution. A test log is the status of the test case (pass/fail).
- To Compare actual results with expected result.

Evaluating Exit criteria and Reporting

Evaluating exit criteria is a process defining when to stop testing. It depends on coverage of code, functionality or risk. Basically it also depends on business risk, cost and time and vary from project to project. Exit criteria come into picture, when:

Maximum test cases are executed with certain pass percentage

Bug rate falls below certain level

When we achieve the deadlines

Evaluating exit criteria has the following major tasks:

- To assess if more test are needed or if the exit criteria specified should be changed
- To write a test summary report for stakeholders

Test Closure activities:

ii. Test Management Issues for Systems of Systems
iii. Test Management Issues for Safety Critical Systems
iv. Other Test Management Issues

Test Management Issues for Exploratory Testing:-

1- Session-based test management (SBTM) is a concept for managing exploratory testing. A session is the basic unit of testing work, uninterrupted, and focused on a specific test object with a specific test objective (the test charter).
2- At the end of a single session, a report, typically called a session sheet is produced on the activities performed. SBTM operates within a documented process structure and produces records that complement verification documentation.
3- A test session can be separated into three stages:
   - Session Setup: Setting up the test environment and improving the understanding of the product.
   - Test Design and Execution: Scanning the test object and looking for problems
   - Defect Investigation and Reporting: Begins when the tester finds something that looks to be a failure.

Test Management Issues for Systems of Systems:-

1- The following issues are associated with the test management of systems of systems:

- Test management is more complex because the testing of the individual systems making up the systems of systems may be conducted at different locations, by different organizations and using different lifecycle models. For these reasons the master test plan for the systems of systems typically implements a formal lifecycle model with emphasis on management issues such as milestones and quality gates. There is often a formally defined Quality Assurance process which may be defined in a separate quality plan.
- Supporting processes such as configuration management, change management and release management must be formally defined and interfaces to test management agreed. These processes are essential to ensure that software deliveries are controlled, changes are introduced in a managed way and the software baselines being tested are defined.

- Lays out a fundamental test process, for example, test planning and control, test analysis and design, test implementation and execution, evaluating of test exit criteria and test reporting, and, test closure activities.
- Describes how to evaluate the effectiveness and efficiency of testing, for example, the percentage of defects to be detected (Defect Detection Percentage or DDP) and the relative cost of defects detected in testing as opposed to after release.
- Defines desired quality targets, such as reliability (e.g. measured in term of failure rate) or usability.
- Specifies activities for test process improvement, for example, application of the Test Maturity Model or Test Process Improvement model, or implementation of recommendations from project retrospectives.
- The test policy may address test activities for new development as well as for maintenance. It may also reference a standard for testing terminology to be used throughout the organization.

Test Strategy :-

1-The test strategy describes the organization's methods of testing, including product and project risk management, the division of testing into levels, or phases, and the high-level activities associated with testing.

2-The test strategy, and the process and activities described in it, should be consistent with the test policy. It should provide the generic test requirements for the organization or for one or more projects.

3- As described in the Foundation Syllabus, test strategies (also called test approaches) may be classified based on when test design begins:

- Preventative strategies design tests early to prevent defects
- Reactive strategies where test design comes after the software or system has been produced.

4- Typical strategies (or approaches) include:

- Analytical strategies, such as risk-based testing
- Model-based strategies, such as operational profiling
- Methodical strategies, such as quality-characteristic based
- Process- or standard-compliant strategies, such as IEEE 829-based
- Dynamic and heuristic strategies, such as using bug-based attacks
- Consultative strategies, such as user-directed testing
- Regression testing strategies, such as extensive automation.

5- Different strategies may be combined. The specific strategy selected should be appropriate to the organization's needs and means, and organizations may tailor strategies to fit particular operations and projects.
6- In many instances, a test strategy explains the project and product risks and describes how the test process manages these risks. In such instances, the connection between risks and testing should be explicitly explained, as are options for reducing and managing these risks.
7- The test strategy may describe the test levels to be carried out. In such cases, it should give high-level guidance on the entry criteria and exit criteria of each level and the relationships among the levels (e.g., division of test coverage objectives).
8- The test strategy may also describe the following:

- Integration procedures
- Test specification techniques
- Independence of testing (which may vary depending on level)
- Mandatory and optional standards
- Test environments
- Test automation
- Reusability of software work products and test work products
- Re-testing and regression testing
- Test control and reporting
- Test measurements and metrics
- Incident management
- Configuration management approach of testware

Both short and long term test strategies should be defined. This can be done in one or more documents. Different test strategies are suitable for different organizations and projects. For example, where security- or safety-critical applications are involved, a more intensive strategy may be more appropriate than in other cases.

Q10-what is test estimation? state the factors that should be considered in test estimation that can influence cost,effort,and duration of the test activities
Ans:-1- Test Estimation is a management activity which approximates how long a task would take to complete.estimating effort for the test is one of the major and important tasks in test management

2-test estimation: The calculated approximation of a result related to various aspects of testing (e.g., effort spent, completion date, costs involved, number of

	partition at least once.	
3	Example:- ○ A text field permits only numeric characters ○ Length must be 6-10 characters long	Example:- an Address text box which allows maximum 500 characters. So, writing test cases for each character once will be very difficult so that will choose boundary value analysis.
4	EPA is not a part of stress and negative testing.	Boundary value analysis is part of stress and negative testing.
5		

Q4-Explain defect taxonomies and their use with example
Ans:-1- defect taxonomy: A system of (hierarchical) categories designed to be a useful aid for reproducibly classifying defects.
2-defect taxonomies is a type of Defect-based techniques.
3-The tester who uses the taxonomy samples from the list, selecting a potential problem for analysis. Taxonomies can list root cause, defect and failure. Defect taxonomies list most common defects in the software under test. The list is used to design test cases.
4-Example:-

```
1  #include <stdio.h>
2  main ()
3  {
4     int i, n, f;
5     printf ("n = ");
6     scanf ("%d", &n);
7     if (n < 0)  {
8        printf ("Invalid: %d\n", n);
9        n = -1;
10    } else  {
11       f = 1;
12       for (i = 1; i <= n; i++) {
13          f *= i;
14       }
15       printf("d! = %d\n", n, f);
16    }
17    return n;
18 }
```

Test Values:
n == 0 (white arrows)

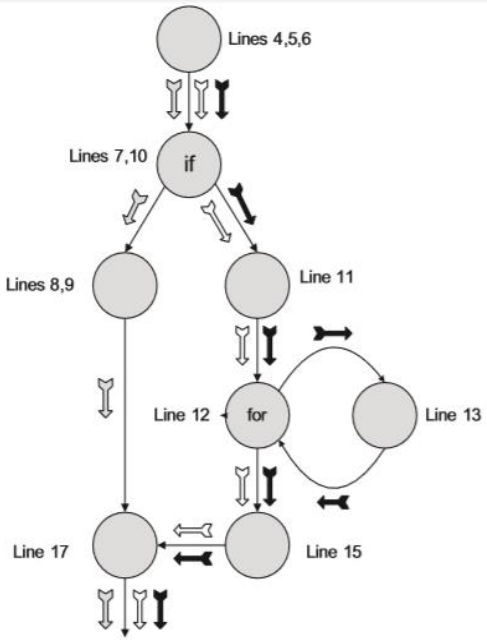


Figure 4–37 *Decision coverage example*

4-The decision statements, such as if/else, switch/case, for, and while statements, are identified.

5- Coverage Criteria:- Number of decision outcomes executed / total number of decision outcomes

Q6-what is error guessing?what is exploratory testing explain

Ans:- 1-error guessing: A test design technique where the experience of the tester is used to anticipate what defects might be present in the component or system under test as a result of errors made and to design tests specifically to expose them.

2-The tester uses experience to guess the potential errors that might have been made and determines the methods to uncover the resulting defects. Error guessing is also useful during risk analysis to identify potential failure modes.

3- error guessing involves the tester taking guesses about a mistake that a programmer might make and then developing tests for it. Notice that this is what might be a called a “gray-box” test since it requires the tester to have some idea about typical programming mistakes, how those mistakes become

UNIT-4

Q1-what is domain testing? Explain accuracy and usability as quality attribute for domain testing?

Ans:-1-Domain testing is one of the most widely practiced software testing techniques.it is a method of selecting a small number of test cases from a nearly infinite group of candidate test cases.

2-Quality attributes for domain testing:

- Functional testing is focused on "what" the product does. The test basis for functional testing is generally a requirements or specification document, specific domain expertise or implied need. Functional tests vary according to the test level or phase in which they are conducted. For example, a functional test conducted during integration testing will test the functionality of interfacing modules which implement a single defined function. At the system test level, functional tests include testing the functionality of the application as a whole. For systems of systems, functional testing will focus primarily on end to end testing across the integrated systems.
- Functional testing may be performed by a dedicated tester, a domain expert, or a developer (usually at the component level).
- The following quality attributes are considered:
  i. Accuracy
  ii. Suitability
  iii. Interoperability
  iv. Functional security
  v. Usability
  vi. Accessibility

3-ACCURACY:

- accuracy: The capability of the software product to provide the right or agreed-upon results or effects with the needed degree of precision. accuracy testing: The process of testing to determine the accuracy of a software product.
- A good synonym for accuracy is correctness.

- Suitability testing is focused on the appropriateness of a set of functions relative to its intended, specific tasks. In other words, given the problems we need to solve, can the system solve them?
- Like accuracy, this particular quality characteristic will be shared with the test analysts. We would expect that we would share use cases and scenarios and discover useful test oracles with them.
- Static testing at the low-level design and code phases would be predominately done by technical testers with domain testers being more involved in the requirements and high-level design phases.

Interoperability:

- interoperability: The capability of the software product to interact with one or more specified components or systems.
  interoperability testing: The process of testing to determine the interoperability of a software product.
- Interoperability is defined as the capability of the system to interact with one or more specified systems in all target environments.
- Test analysts will be concerned with the end-to-end testing of scenarios using use cases, pairwise testing, classification trees, and other techniques.
- Interoperability testing for technical test analysts will include ensuring the smooth functioning of data transfers between systems and checking that the interfaces between systems integrate smoothly. This testing is most visible during system integration and systems testing, but clearly the foundation will have to be built before that.

Usability:

- usability: The capability of the software to be understood, learned, used, and attractive to the user when used under specified conditions.
  usability testing: Testing to determine the extent to which the software product is understood, easy to learn, easy to operate, and attractive to the users under specified conditions.
- Usability testing, naturally enough, focuses on the users. This is why many notable usability experts and usability test experts have a background in psychology rather than simply being technologists or domain experts

7- As you can see, this is a very detailed code review checklist. However, customization based on your own experience, and your organization's needs, is encouraged.

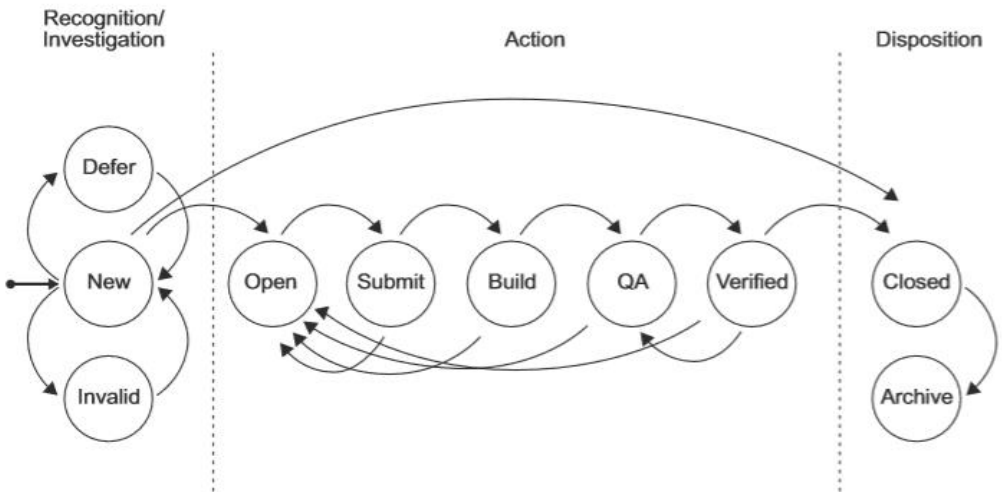
Q7- Explain the incident management lifecycle.

Ans:- 1- Incident management is a term describing the activities of an organization to identify, analyze, and correct hazards to prevent a future re-occurrence. If not managed, an incident can escalate into an emergency, crisis or a disaster.

2-These incidents within a structured organization are normally dealt with by either an incident response team (IRT), an incident management team (IMT), or Incident Command System (ICS). Without effective incident management, an incident can disrupt business operations, information security, IT systems, employees, customers, or other vital business functions.

3-An incident is an event that could lead to loss of, or disruption to, an organization's operations, services or functions.

4-Incident management is therefore the process of limiting the potential disruption caused by such an event, followed by a return to business as usual. Without effective incident management, an incident can disrupt business operations, information security, IT systems, employees, customers, or other vital business functions.



Q8-Explain the process for applying the IEEE 1044 standard to your organization

5-Design problems. Conceptual problems. Standards and guidelines (or lack of same) compliance. We have a good way of finding these kinds of issues. It's called static testing. From the first requirements to the latest patch, there is likely no better way to ensure that the system is maintainable.

Q13-what are the different roles during formal review process? What are their responsibility? explain

Ans:-1-A formal review is one of the important review techniques in static testing.it has a complete process to follow.unlike the informal review,formal review is more structured and regulated.

2-During a formal review, there are some essential roles and responsibilities:

■ The manager. The manager allocates resources, schedules reviews, and the like. However, the manager might not be allowed to attend based on the review type.

■ The moderator or leader: This is the chair of the review meeting.

■ The author: This is the person who wrote the item under review. A review meeting, done properly, should not be a sad or humiliating experience for the author.

■ The reviewers. These are the people who examine the item under review, possibly finding defects in it. Reviewers can play specialized roles based on their expertise or based on some type of defect they should target.

■ The scribe or secretary or recorder: This is the person who writes down the findings.