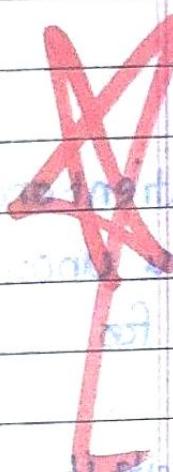


6) Testing Tools and Measurements

Page No.	88
Date	

6.1 Limitations of manual testing and need for automatic testing tools:

Manual testing is the process of manually testing software for defects. It requires a tester to play role of an end user and use most of all applications to ensure correct behaviour.



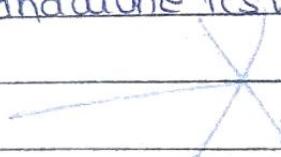
Limitations of manual testing:-

- * Time consuming process
- * Limited support for regression testing
- * Error prone testing
- * Impractical performance testing.
- * Non-consistent effort.
- * Limited scope it is very difficult
- * No batch testing at global level

Automation testing is a process to reduce manual testing work by using tools and strategies that reduce human involvement or interaction in unskilled, repetitive or redundant tasks.

The automation testing process includes:-

- * Detailed test cases
- * standalone test environment



Benefits of automation testing :-

- * Reduce time of testing.
- * Improve the bugs finding.
- * Delivers the quality product.
- * Allow to run tests many times with different data.
- * Getting more time for test planning.
- * save resources or requires less resources.
- * Automation never tires, an expert person can work at a time many tools.

General Approaches for automation test :-

- * Code driven testing :- This approach is used to test public interfaces to objects , classes, modules or libraries are tested using different types of inputs to validate the result with return results.
- * Graphical user interface testing :- This approach is used testing frameworks for generating testing different user interface events like keystrokes and mouse clicks and notices change in that user interface.
- * API driven testing :- A testing framework that uses programming interface of an application to validate the behaviour of application under test.

✓ Need for automated testing tools

- * Using testing effectively : Automation of testing allows machines to complete tedious, repetitive work while human personnel perform other tasks.

An automated test executes the operation in machine speed, allowing test to be completed in efficient and faster way.

Advantages of automated testing

- * Reducing test costs : The testing process using automated methods require only a fraction of the computer hardware that would be necessary to complete a manual test.

- * Replicating testing across different platforms : When applications need to deployed across different hardware or software platforms, standard or benchmark tests can be created and repeated on target platforms to ensure that new platform operates consistently.

* Greater application coverage

- * Greater application test coverage individually reduces the risk if exposing user to malfunctioning or non-compliant software.

* Results reports: Full featured automated testing produce convenient test reporting and analysis.

6.2

Features of Test tool

Guidelines for Static Testing Tools

Static analysis tools are generally used by developers as a part of development and component testing process. The key aspect is that the code is not executed or run but the tool itself is executed, and the source code we are interested in is the input data to the tool.

These tools are mostly used by developers.

⇒ Features / characteristics of static analysis tool are :-

- * To calculate metrics such as cyclomatic complexity or nesting levels.

- * To enforce coding standards

- * To analyse structure and dependencies.

- * Help in code understanding

- * To identify anomalies or defects in the code.

Guidelines for Dynamic Testing Tools :-

These are dynamic because they require the code to be in running state. They do analysis rather than testing tools because they analyse what is happening in the code while software is running.

→ Features or characteristics of dynamic analysis

tools are as follows:-

- * To detect memory leaks
- * To identify pointer arithmetic errors such as null pointers
- * To identify time dependencies.

~~Testing tools can be classified into following two categories:-~~

① Static Test tools :- These tools do not involve actual input or output. Rather they take a symbolic approach to testing i.e. they do not test the actual execution of software. These tools include the following:-

* Flow analysers :- They ensure consistency in data flow from input to output

* Path tests :- It ensures that all logic paths are tested.

* Coverage analysers :- They find unused or redundant code and code with contradictions.

* Interface analysers :- It examines the effect of passing variables and data between modules.

Q. What are static analysis of software?

A. Static analysis of software does not control

the execution of program. It analyses the

~~②~~ Dynamic Test tools:- These tools test the software system with 'live' data. Dynamic test tools include the following :-

* Test drivers :- It inputs data into a module - under-test (UUT).

* Test beds :- It simultaneously displays source code along with program under execution.

* Emulators :- The response facilities are used to emulate parts of the system not yet developed or glad library has not yet

* Mutation analysis :- The errors are deliberately introduced into the code in order to test the fault tolerance of the system.

~~6.3 Advantages & Disadvantages of Using Tools :-~~

~~Advantages of using tools :-~~

* Reduction of repetitive work.

* Greater consistency and repeatability.

* Objective assessment.

* Ease of access to information about tests or testing.

~~Disadvantages of using tools :-~~

* Unrealistic expectations from tool.

* People often make mistakes by underestimating the time, cost and effort for the initial introduction of the tool.

* People frequently miscalculate the time and effort needed to achieve significant and continuing benefits from the tool.

- * Mostly people underestimate the effort required to maintain the test assets generated by the tool.
- * People depend on tool a lot.

6.4 Selecting a Testing tool

While introducing the tool in the organisation, it must match a need within the organisation and solve that need in a way that is both effective and efficient.

The tool should help in building the strengths of the organisation and should also address its weaknesses. The organisation needs to be ready for the changes that will come along with the new tool.

The following factors are important during tool selection.

- * Assessment of organisation's maturity
- * Identification of areas within the organisation where tool support will help to improve testing process.
- * Evaluation of tools against clear requirements and objective criteria.
- * Proof of concept to see whether the product works as desired and meets the requirements and objectives defined for it.
- * Evaluation of vendor or open source network support.
- * Identifying and planning internal implementation.

6.5

Automated test tools

Automated tests are suitable for following purposes:

- * Regression testing for a stable system that will be run on regular basis.

- * Fast data creation in test systems where the database must be wiped off on regular basis.

Automated tests are not suitable for the following purposes:

- * Testing new functionality.

- * Regression testing systems that are expected to have significant user interface changes.

Testing using Automated tools (Test automation) :-

Software test automation makes use of specialized tools to control the execution of tests and compares the actual result against the expected result.

Criteria for tool selection:-

- * Data driven capabilities.

- * Debugging and logging capabilities.

- * Platform independence.

- * Extensibility and customizability.

- * Email notifications.

- * Version control friendly.

- * Support unattended test runs.

Types of frameworks:

Typically there are 4 test automation frameworks that are adopted while automating the applications.

- * Data driven automation framework

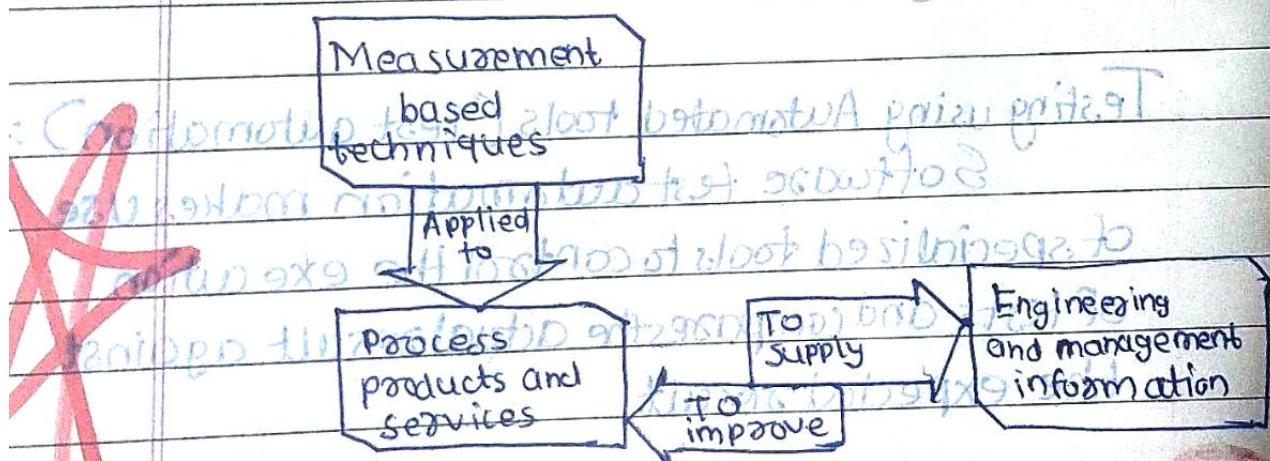
- * Keyword driven automation framework

- * Modular automation framework

- * Hybrid automation framework.

6.6) Metrics and measurement of software testing:

Measurement plays a critical role in effective and efficient software development, making measurements of the software development is avoid of bugs & errors.



Metrics:

Metric is a standard unit of measurement that quantifies result. They are used for evaluating the software processes, products and services is termed as software metrics.

~~Importance of metrics~~ ~~what do we mean by metrics~~

- * Metrics are used to improve quality and productivity of products and services thus achieving customer satisfaction.
- * Easy for management to digest one number and drill down, if required.
- * Different metric trend act as monitor when the process is going of out of control.
- * They provide improvement for current process.
- * Take decision for next phase of activities.
- * Evidence of the claim or prediction.
- * Understand the type of improvement required.
- * Take decision on process or technology change.

~~Types of metrics~~

~~Testing metrics~~ Testing metrics are divided into two categories :-

- * Base metrics
- * calculated metrics

~~Base metrics~~ These are the metrics which are derived from the data gathered by Test Analyst during the test case development and execution.

This data will be tracked throughout the test life cycle i.e. collecting data like, Total no. of test cases developed for project or no. of test cases need to be executed or no. of test cases passed / failed / block.

Calculated metrics

Calculated metrics are derived from the data gathered in base metrics. These metrics are generally tracked down by test lead / manager for Test Reporting purpose.

Definitions and formulas for calculating metrics

- 1] % of test cases executed :- This metric is used to obtain the execution status of the test cases in terms of percentage.

Red

$$\% \text{ of test cases} = \frac{\text{No. of test cases executed}}{\text{executed}} \times 100$$

.....

- 2] % of test case not executed :- This metric is used to obtain pending execution status of the test cases in terms of percentage.

$$\% \text{ of test cases} = \frac{\text{No. of test cases not executed}}{\text{not executed}} \times 100$$

.....

.....

.....

.....

.....

.....

~~Project metrics~~ → ~~are also called~~ ~~indicators~~

Project metrics enable software project managers to :- ~~to make better decisions~~

- * Assess the status of ongoing project

- * Track potential risks in ~~in advance~~

- * Uncover problem areas before their status becomes critical

- * Adjust work flow or tasks

- * Evaluate project team's ability to control quality of software ~~work products~~

- * Project metrics are used for making tactical decisions → They are used to adopt project workflow and technical activities

Use of project metrics → ~~to facilitate~~

The first application of project metrics occurs during estimation. Metrics from past projects are used as a basis for estimating time and effort.

As the project proceeds, the amount of time and effort expended are compared to original estimates.

As technical work commences, other project metrics become important. Production rates and errors uncovered during each generic framework activity are measured.

→ ~~initial (at 77 to 1977) = virtual~~

Project metrics are used to ~~control~~ monitor the project

- * Minimise the development schedule by making the adjustment necessary to avoid delays and mitigate potential problems and risks.

- * Assess product quality on an ongoing basis and when necessary modify the technical approach to improve quality.

~~Test coverage is not enough~~

Progress metrics - How to follow

Progress metric is the set of metrics to indicate how different activities of the project are progressing.

Usually when we are tracking progress it is related to time, or other unit that indicates a schedule.

If we are financial people, then we track money spent. But for software quality assurance we track progress of things as defects, test cases, man hours etc.

~~It can also be bugs found & submitted~~

Productivity metrics:-

Productivity metric is the metric that is collected by considering various other productivity numbers. This can be effectively used for activities like planning and tracking.

$$\text{Productivity} = (\text{Planned Effort}) / (\text{Actual Effort})$$

*8