

CO5 :

Test software for performance measure using automation testing tools.

Manual Testing

- Manual testing is a testing process that is carried out manually in order to find defects without the usage of tools or automation scripting.
- A test plan document is prepared that acts as a guide to the testing process in order to have the complete test coverage.

How to Do Manual Testing

- *Requirement Analysis*
- *Test Plan Creation*
- *Test case Creation*
- *Test case Execution*
- *Defect Logging*
- *Defect Fix & Re-Verification*



Limitations of Manual Testing

i) Manual Testing requires more time or more resources, sometimes both Time and Resources.

Covering all areas of the Application requires more Tests, Creating all possible Test cases, and executing Test cases takes more time. If it is Test Automation, Test tool can execute Tests quickly.

ii) Less Accuracy

Human Users (Testers) may make mistakes, so we cannot expect more accuracy in Manual Testing,

If it is Test Automation / Automated Testing, if you provide the correct logic then test tool can provide correct output.

iii) Performance testing is impractical in manual testing.

Unit 5: Testing Tools and Measurements

Organizing Thousands of Machines / Computers and human Users is impractical, If it is Test Automation, we can create thousands of Virtual users and using 3 or 4 Computers we can apply the Load and test the Performance of the Application

iv) Comparing large amount of data is impractical.

Comparing two Databases that have thousands of records is impractical, but it is very is in Test Automation.

v) Processing change requests during software maintenance takes more time

vi) Batch Testing is possible, but for each test execution Human user interaction is mandatory.

Batch Testing means executing series of tests, In Batch Testing for every test case execution User / Tester interaction is mandatory, If it is Test Automation Test tool can execute series of Tests without human user interaction.

vii) GUI Objects Size difference and Color combinations etc.. are not easy to find in Manual Testing.

viii) Manual Test Case scope is very less, if it is automated test, scope is more.

In Manual Testing, Test case scope is very limited why because Tester/user can concentrate on one or two Verification points only, If it is Test Automation, Test tool (Tool also Software) can concentration on multiple verification points at a time.

ix) Executing same tests again and again is time taking process as well as Tedious.

Sometimes we need to execute same tests using multiple sets of Test data, for each test iteration user interaction is mandatory, In Test Automation using Test Data, data file (either Text file or Excel file or Database file) we can easily conduct Data driven Testing.

x) For every release you must rerun the same set of tests which can be tiresome.

We need to execute Sanity Test Cases and Regression Test cases on every modified build, it takes more time. In Automated Testing / Test Automation once we can create Tests then Tool can execute Tests multiple times quickly.

Automation Testing

- Automation testing is a technique uses an application to implement entire life cycle of the software in less time and provides efficiency and effectiveness to the testing software.
- Automation testing is an Automatic technique where the tester writes scripts by own and uses suitable software to test the software.

Unit 5: Testing Tools and Measurements

- It is basically an automation process of a manual process.
- The main goal of Automation testing is to increase the test efficiency and develop software value.

Test automation is the use of special software to control the execution of tests and the comparison of actual outcomes with predicted outcomes. The objective of automated testing is to simplify as much of the testing effort as possible with a minimum set of scripts. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually.

Types of test automation tools:

- **Static automation tools:** These tools are used throughout a software development lifecycle, e.g. tools used for verification purposes. There are many varieties of static testing tools used by different people as per the type of system being developed. These tools do not involve actual input and output. Rather, they take a symbolic approach to testing, i.e. they do not test the actual execution of the software. e.g. Flow analyzers, Coverage analyzers, Interface analyzer Code complexity measurement tools can be used to measure the complexity of a given code. Similarly, data-profiling tools can be used to optimize a database. Code-profiling tools can be used to optimize code. Test-generators are used for generating a test plan from code. Syntax-checking tools are used to verify correctness of code.
- **Dynamic automation tools:** These tools test the software system with live data. They are used at different levels of testing starting from unit testing & which may go up to system testing & performance testing. These tools are generally used by tester. These tools test the software system with live data. e.g. Test driver, Test beds, Emulators There are many different tools used for dynamic testing. Some of the areas covered by testing tools are:
 1. Regression testing using automated tools.
 2. Defect tracking and communication systems used by tracking & communication. Performance, Load, stress-testing tools

Benefits of Automation Testing

- Reduces time of testing
- Improve the bugs finding
- Deliver the quality product
- Allow to run tests many time with different data
- Getting more time for test planning
- Save resources or requires less
- Automation never tires, and expert person can work at a time many tools.

Advantages of Switching To Automated Testing from Manual Testing

- Efficient testing

Unit 5: Testing Tools and Measurements

- Consistency in testing
- Better quality software
- Automated testing is cheaper
- Automation testing is faster
- Automated testing is more reliable
- Automated testing reduces human and technical risks
- Automated testing is more powerful and versatile

Features of automated testing tools

- FAST Automation Engine
- Object Eye Internal Recorder
- Visual Recorder
- Multiple Browsers Support
- Dynamic Test Data Support
- Continuous Server Integration
- Mobile Testing Support
- Robust Reporting & Logs
- Reusable Methods
- Integration with Bug Tracking tools
- Integration with Test Management Tools
- Job Scheduler
- Image Comparison
- Distributed Test Execution
- Captcha Automation
- Risk Based Testing

Static Testing Tool

- Static testing tools are used during static analysis of a system.
- Static testing tools are used throughout a software development life cycle, e.g. , tools used for verification purposes.
- There are many varieties of static testing tools used by different people as per the type of system being developed.
- Code complexity measurement tools can be used to measure the complexity of a given code.
- Similarly, data-profiling tools can be used to optimize a database.
- Code-profiling tools can be used to optimize code.
- Test-generators are used for generating a test plan from code.
- Syntax-checking tools are used to verify correctness of code.

Unit 5: Testing Tools and Measurements

Features for selecting static test tools:

- i. Assessment of the organization's maturity (e.g. readiness for change);
- ii. Identification of the areas within the organization where tool support will help to improve testing processes;
- iii. Evaluation of tools against clear requirements and objective criteria;
- iv. Proof-of-concept to see whether the product works as desired and meets the requirements and objectives defined for it;
- v. Evaluation of the vendor (training, support and other commercial aspects) or open-source network of support;
- vi. Identifying and planning internal implementation (including coaching and mentoring for those new to the use of the tool).
 - Static test tools includes:
 1. Flow analyzer :ensures consistency in data flow from input to output
 2. Path tests :finds unused codes and codes with contradictions
 3. Coverage analyzer :all logical paths are tested
 4. Interface analyzer :examines effects of passing variables and data between modules

Dynamic Testing Tool

- Dynamic testing tools are used at different levels of testing starting from unit testing & which may go up to system testing & performance testing.
- These tools are generally used by tester.
- There are many different tools used for dynamic testing. Some of the areas covered by testing tools are:
 1. Regression testing using automated tools.
 2. Defect tracking and communication systems used by tracking & communication.
- Performance, Load, stress-testing tools.

Features for selecting dynamic test tools:

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

Dynamic test tools includes:

1. Test driver :includes data into module under test (MUT)
2. Test beds :simultaneously displays source code along with the program under execution
3. Emulators

Unit 5: Testing Tools and Measurements

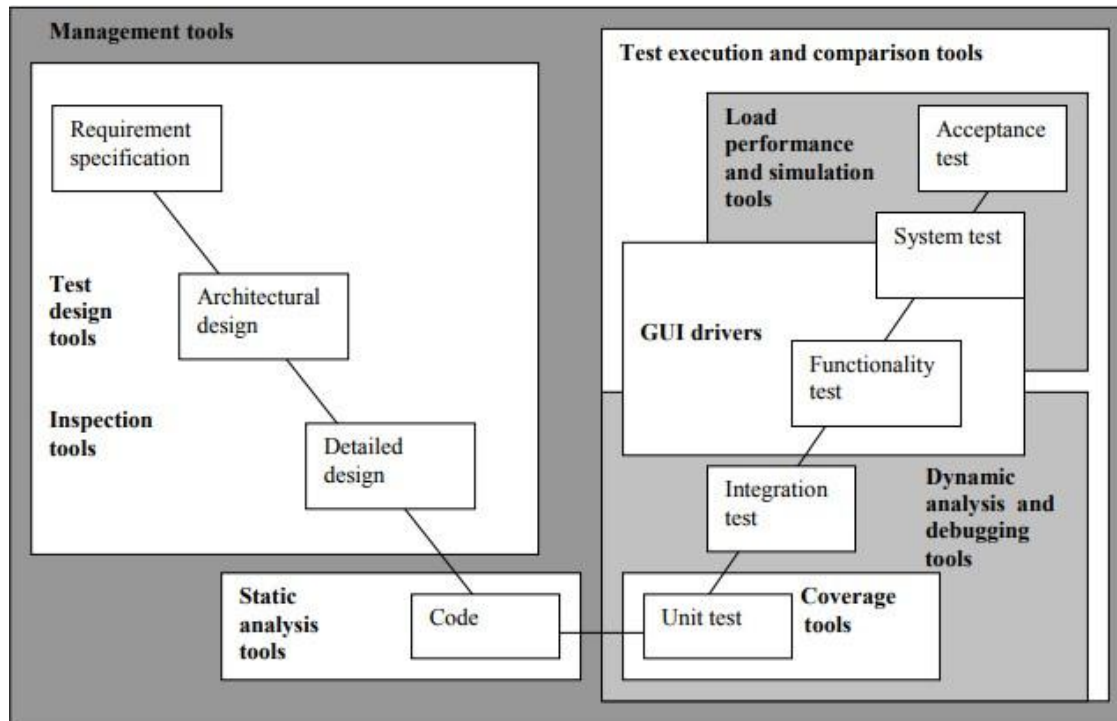


Figure 2. Division of the tools in the software development life cycle (V-model)

Advantages using Tools

- 1. Speed.** The automation tools tests the software under tests with the very faster speed. There is a vast difference between the speed of user entering the data and the automated tools generating and entering the data required for the testing of the software. Speed of this software also completes the work faster.
- 2. Efficiency.** While testers are busy running test cases, testers can't be doing anything else. If the tester have a test tool that reduces the time it takes for him to run his tests, he has more time for test planning and thinking up new tests.
- 3. Accuracy and Precision.** A test tool will perform the same test and check the results perfectly, each and every time.
- 4. Resource Reduction.** Sometimes it can be physically impossible to perform a certain test case. The number of people or the amount of equipment required to create the test condition could be prohibitive. A test tool can be used to simulate the real world and greatly reduce the physical resources necessary to perform the testing.
- 5. Simulation and Emulation.** Test tools are often used to replace hardware or software that would normally interface to your product. This "fake" device or application can then be used to drive or respond to your software in ways that you choose and ways that might otherwise be difficult to achieve.
- 6. Relentlessness.** Test tools and automation never tire or give up. They can keep going and going and on and on without any problem; whereas the tester gets tired to test again and again.

Unit 5: Testing Tools and Measurements

OR

1. Reduce time of testing
2. Improve the bugs finding
3. Deliver the quality software/product
4. Allow to run tests many times with different data
5. Getting more time for test planning
6. Save resources or reduce requirement
7. It is never tired and expert person can work at a time many tools.

Disadvantages of using tools:

- Unrealistic expectation from the tool
- People always make mistake by understanding 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 tools
- Mostly people underestimate the effort required to maintain the test assets generated by the tool
- People depend on the tool a lot.(over reliance on the tool)

Guidelines for selecting a tool:

1. The tool must match its intended use. Wrong selection of a tool can lead to problems like lower efficiency and effectiveness of testing may be lost.
2. Different phases of a life cycle have different quality-factor requirements. Tools required at each stage may differ significantly.
3. Matching a tool with the skills of testers is also essential. If the testers do not have proper training and skill then they may not be able to work effectively.
4. Select affordable tools. Cost and benefits of various tools must be compared before making final decision.
5. Backdoor entry of tools must be prevented. Unauthorized entry results into failure of tool and creates a negative environment for new tool introduction.

Criteria for Selecting Test Tools:

- The Criteria's for selecting Test Tools are,
 1. Meeting requirements;
 2. Technology expectations;
 3. Training/skills;
 4. Management aspects.

Unit 5: Testing Tools and Measurements

1. Meeting requirements

There are plenty of tools available in the market but rarely do they meet all the requirements of a given product or a given organization. Evaluating different tools for different requirements involve significant effort, money, and time. Given of the plethora of choice available, huge delay is involved in selecting and implementing test tools.

2. Technology expectations

Test tools in general may not allow test developers to extends/modify the functionality of the framework. So extending the functionality requires going back to the tool vendor and involves additional cost and effort. A good number of test tools require their libraries to be linked with product binaries

3. Training/skills

While test tools require plenty of training, very few vendors provide the training to the required level. Organization level training is needed to deploy the test tools, as the user of the test suite are not only the test team but also the development team and other areas like configuration management.

4. Management aspects

A test tool increases the system requirement and requires the hardware and software to be upgraded. This increases the cost of the already- expensive test tool.

When to use automated test tools

- **Stress, reliability, scalability and performance testing:**
These types of testing require the test case to be run from a large number of different machines for an extended period of time, such as 24 hours, 48 hours, and so on. It is just not possible to have hundreds of users trying out the product they may be not willing to perform the repetitive tasks, nor will it be possible to find that many people with the required skill sets. Test cases belonging to these testing types become the first candidates for automation.
- **Regression tests:** Regression tests are repetitive in nature .These test cases are executed multiple times during the product development phase. Given the repetitive nature of test cases, automation will save significant time and effort in the long run. The time thus gained can be effectively utilized for other tests.
- **Functional tests:** These kinds of tests may require a complex set up and thus require specialized skill, which may not be available on an ongoing basis. Automating these once, using the expert skill sets, can enable using less-skilled people to run these test on an ongoing basis

Metrics and measurements

Metrics & measurement:

Metrics is a relative measurement of status of process or product in terms of two or more entities taken together for comparison.

Measurements are key element for controlling software engineering processes.

Need of software measurements:

1. Understanding: Metrics can help in making the aspects of process more visible, thereby giving a better understanding of the relationship among the activities and entities they affect.
2. Control: Using baselines, goals and an understanding of the relationships, we can predict what is likely to happen and correspondingly, make appropriate changes in the process to help meet the goals.
3. Improvement: By taking corrective actions and making appropriate changes, we can improve a product. Similarly, based on the analysis of a project, a process can also be improved.

Metrics classification

Metrics are basically classified as:

1. Product Metrics: Product metrics are measures of software product at any stage of its development, from requirements to installed system.
2. Process Metrics: Process metrics are measures of the software development process such as the overall development time, type of methodology used or the average level of experience of the programming staff.

Product Metrics is classified as

1. **Project Metrics:** A set of metrics that indicates how the project is planned and executed.
2. **Progress:** A set of metrics that tracks how the different activities of the project are progressing.

Progress Metrics is classified as

1. Test defect metrics : help the testing team in analysis of product quality and testing.
 2. Development defect metrics: help the development team in analysis of development activities.
3. **Productivity:** A set of metrics that takes into account various productivity numbers that can be collected and used for planning and tracking testing activities.