# Testing Manners and Technologies: An Analysis

Arpita Tewari
Research Scholar,
Uttar Pradesh Rajarshi Tandon Open University,
Allahabad

A.K.Misra
Professor,
Department of Computer Science and Engineering, MNNIT, Allahabad

## ABSTRACT
In the present paper traditional software testing manners and technologies have been compared with the vibrant trends. The paper also discusses about the testing manners of today's trends that have been also changed due to proliferation of software and rapid change. Some complications in testing have also been highlighted and requirement of adopting advanced degree automation techniques have been discussed. The paper also describes different phases of testing techniques and its future prospects to make the testing a perfect remedy for a difficult and intractable problem.

**Keywords:** software testing, software models, test selection techniques

## I. INTRODUCTION
Software test "a test to verify if there is no problem with software"[1]; commonly known as a test to evaluate and then guarantee the level of quality of software ".Furthermore, software testing is defined as a process of quality assurance or verification".[8] Classical definition of Myers says that software testing is the process of executing a program that intent to find errors[2],while the vibrant trend says software testing is the process used to identify the correctness, completeness and the quality of the developed computer software.[2] According to the standard which stipulates the life cycle of software, JJIS X 0160:1996(software lifecycle process), the test is broken into three phases:

1. *System and software qualification confirmation tests for the development process.*
2. *Operation tests for the operation process.*
3. *Quality assurance process, verification process and validation process that include testing as part of the lifecycle supporting processes* [1].

According to JIS X 0170, which includes not only software but also hardware systems, addresses a task called software testing in the processes of verification and validation.

Besides assessing the software quality, the software testing is also used to review the specification, design and coding of software.

In other words a group or set of activities related to affirming the quality of a program that can be planned in advance and performed systematically exercising the code. ISO 9126 defines six high- level quality characteristics that can be used in software evaluation. *It includes functionality, reliability, usability, efficiency, maintainability, and portability* [9].

## II. OBJECTIVE OF TESTING
The primary objective of testing is:

1. Whether the application is working as expected without any errors or bugs.

2. Whether the performance of the application is as expected & meets the need.

Due to the broad area of software testing that involves many technical and non-technical areas the definition of testing varies according to the purpose, process, and level of testing described. A good description of Miller that Software testing aim is to affirm the quality of software systems by systematically exercising the software in carefully controlled circumstances.

Offshore testing, one of the leading destinations is globally accepted. It not only brings objectivity and transparency to defect reporting process, but also improves the core business strategy. In the present scenario, testing single goal is to uncover faults by triggering failures.[2] [6]

The standards of progress include:

1. Degree of acceptance
2. Degree of dependability
3. Change of research paradigms
4. Feasibility of techniques.

## III. THE TESTING SPECTRUM
Code coverage Analysis in structural testing technique (AKA glass box testing & white box testing) behaves against the apparent intention of the source code. This is in contrast with the functional testing (AKA black box testing), which compares test program behaviour against requirement specification.

Structural testing (also called path testing) examines how the programs works, taking into account possible pitfalls in the structure logic. Functional testing examines what the program accomplishes, without regarded to how it works internally.

At first glance, structural testing seems unsafe because it could not find errors of omission. However, requirement specifications sometimes do not exist, & are rarely complete. This is especially true near the end of the product development time line when the requirement specifications is updated less frequently and the product itself begins to take over the role of the specification. The difference between functional & structural testing blurs near release line.

## IV. TESTING ACTIONS AND DEVELOPMENT STAGE

Conventional and modern model of development and test has been presented in the different manner in which the test is conducted after software development; consequently, failures are detected and corrected only after software development. Defects in the upper stream cannot be remedied until the end point of testing, which led to an increase of workload for correction. To overcome the problem, the recent model of w shaped testing and development has been introduced.

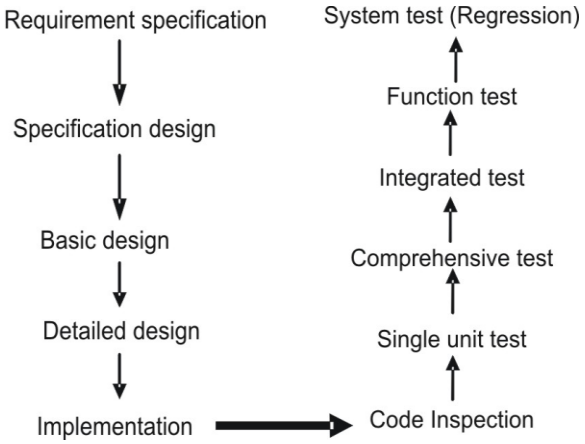



**Fig.1 The most recent model of software development and tests**

## V. TAXONOMY OF TESTING TECHNIQUES

Software testing is a very broad area, which involves many other technical and non-technical areas, such as specification, design and implementation, maintenance, process and management issues in software engineering. Our study focuses on the state of art; the art of testing techniques.[6] Research and development activities are

**Stress testing:** It is a form of testing that is used to determine the stability of a given system or entity &

**Operations Testing Technique:** Operations testing is designed to verify prior to production that the operating procedures and staff can properly execute the application.[14]

**Compliance Testing Technique:** System is developed in accordance with standards and procedures.

**Regression Testing techniques:** Regression testing is an expensive and frequently executed maintenance process used to revalidate modified software. To improve it, regression test selection (RTS) techniques strive to lower costs without verily reducing effectiveness by carefully selecting a subset of the test

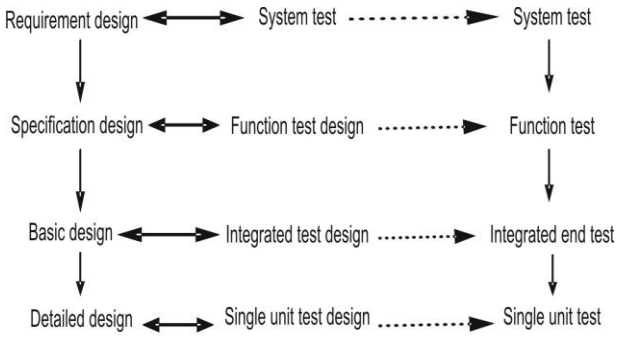



**Fig.2 The conventional model of Development and test**

Involves testing beyond normal operational capacity. It determines that system performs with expected volumes.

**Execution Testing Technique:** This technique determines whether the system achieves the desired level of proficiency in a production status; can verify response times, turnaround times and can be tested in whole or impair using the actual system or a simulated model of a system.

**Recovery Testing Technique:** It is the activity of testing that how well the application is able to recover from crashes, hardware failures and similar problems? It can involve the manual functions of an application, loss of input capability, loss of input capability ; loss of communication lines, hardware or operating system failure, loss of database integrity, operator error or application system failure.

suite. Under certain conditions, some can even guarantee that the selected test cases perform no worse than the original test suite. Developers often do *regression testing* to search for such regression errors. The simplest regression testing strategy is to rerun all existing test cases.

**Parallel Testing Techniques:** Old system and new system are run and the results compared to detect unplanned differences.

**Requirement Testing Technique:** it verifies that the system can perform its function correctly and that the correctness can be sustained over a continuous period of time.

Requirement testing is primarily performed through the creation of test conditions and functional checklists.

**Control Testing Technique:** Control, a management tool to reduce system risk to an acceptable level. Controls include data validation, file integrity, audit trail, backup and recovery, documentation, and the other aspects of systems related to integrity.

**Error handling testing technique:** Errors can be prevented or detected, and then corrected. Error handling testing determines the ability of application system to properly process incorrect transactions.

**Inter system testing technique:** Data is correctly passed from system to system and ensures that proper coordination and timing of functions exists between the application systems.

Application systems are frequently interconnected to other application systems.

**Manual Support:** It involves all the functions performed by people in preparing data for, and using data from, automated applications.

**Static testing (Verification technique) : V**erification is the act of reviewing, inspecting, testing, etc. to establish and document that a product, service, or system meets the regulatory, standard, or specification requirements.

**Requirement Reviews:** The study and discussion of the computer system requirements to ensure they meet stated user needs and are feasible.

**Design Reviews:** The study and discussion of the computer system design to ensure it will support the system requirements.

**Code Walkthroughs:** An informal analysis of the program source code to find defects and verify coding techniques

**Code Inspections:** A formal analysis of the program source code to find defects as defined by meeting computer system design specifications[14]

**Dynamic testing (Validation technique): V**alidation refers to the process of data validation, controlling that data inserted into an application satisfies pre determined formats or complies with stated length and character requirements and other defined input criteria[15].

**Unit testing:** it is done at the lowest level. it tests the basic unit of software, which is the smallest testable piece of software, and is often called "unit", "module", or "component" interchangeably.[]

**Integrated testing:** The testing of related programs, modules, or units of code, validates that multiple parts of the system interact according to the system design. Integration testing of object oriented programs is State Based Testing, Event Based Techniques, Testing against Formal Specifications, Deterministic and Reachability Testing Techniques, Fault Based Techniques, UML Based Techniques, Data Flow Analysis.

**System testing:** The testing of an entire computer system, includes functional and structural testing, such as stress testing.

**User acceptance testing:** The testing of a computer system or parts of a computer system to make sure it will work in the system regardless of what the system requirements indicate.

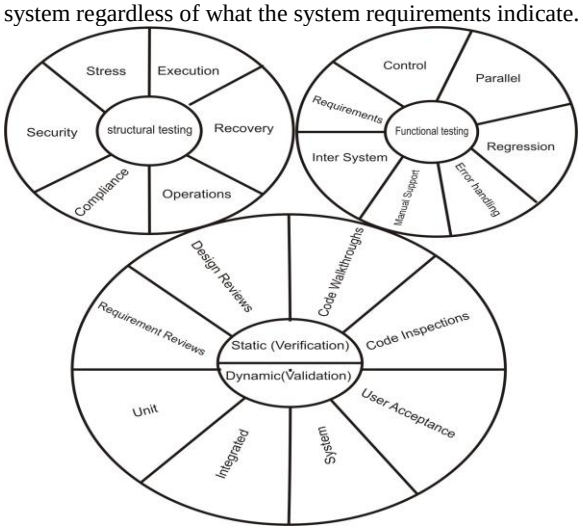



**Fig.3 Phases of Testing**

## VI. REQUIREMENT OF TESTING FOR RAPID CHANGE

Tests are done to certify the code maintenance that did not accidentally insert defects into working parts of the software.

Software testing in current industry (business) trends includes:

1. Web based applications
2. Service oriented Architectures(SOA)
3. S/W as a service (SAAS)
4. Wireless Technologies
5. Mobile Technologies

## VII. TESTING COMPLICATIONS

In the real life challenges scenario; it has been realized that most of the testing complexities are non-technical as "the bewildered testing team" can not focus on how to perform the tasks by an aggressive deadline. "the test maintenance failure", manual testing ,"the uncertainty principal", confusion choosing the right tests[7]

In most common automation technique of testing, design, documentation and maintenance of these tests are time-costly human tasks.

Unless the certain conditions are tested for the defects, the software is considered and treated to be at normal state, or "no-problem" software are caused by insufficiency of its testing.

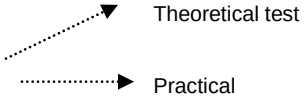
Recent software related problems are largely caused by lack of testing, and have actually brought social problems and financial losses. as an illustration in 2007 there was an inconvenience with the automatic gate system at train stations in and around the Tokyo metropolis; once this condition occurred, it caused errors to all connected automatic gate machines; as this error affected the entire rail industry of the Metropolis area later on the report says that the offset was missing. [1] It was one & only lack of testing that can be of any cause in one of them as

*Lack of training "US" vs. "Them" mentality*
*Lack of testing tools*
*Lack of management understanding /Support of testing*
*Lack of customer and user involvement*
*Not enough time for testing*
*Over reliance on independent testing*
*Rapid change*
*Testers in a "Lose/lose "situation*
*Having to say "No"* [15]

## VIII. PROPOSED TECHNIQUE AND RECOMMENDATION

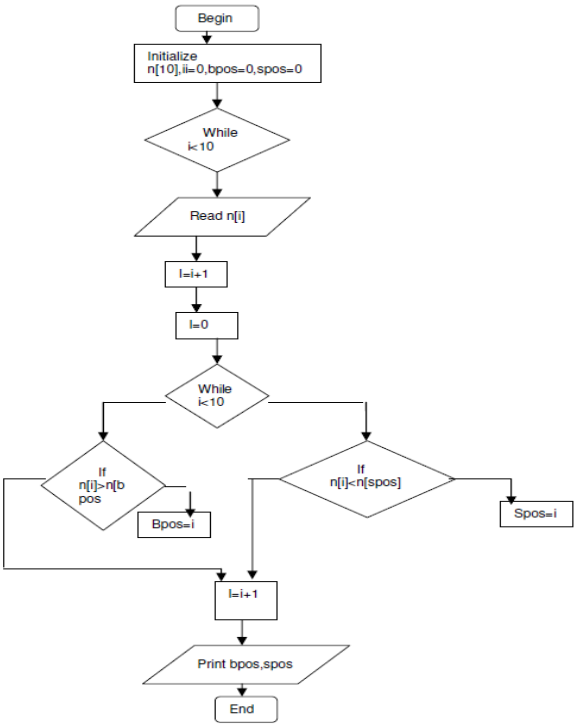
As we can not test everything thoroughly but we must test some things thoroughly and for this testing should rely heavily on high volume automation techniques and to focus on risk and testing, rework is needed. Certain techniques are developed for the solution of that the program will do exactly what it is designed to do. Techniques are developed to construct the formal proof (or disproof) of the correctness of a program.

Proof of correctness (Automatic theorem Proving technique)



To overcome the problem one practical and more intuitive approach in which we attempt to improve our confidence is a program by testing the program for a set of test cases.

## Criteria decided:




"Every statement will be executed at least once during the test"[13] as the problem is designed here to find the location of the biggest and smallest no. in an array having 10 elements n[10].Firstly,I have initialized the value of I as 0 and fix a condition that array may contain only 10 elements these have integer values.Here,we compare the value of n[i] with n[bpos] and if n[i] is greater than n[bpos] then bpos=i and we get the desired point if and only if n[i]>n[bpos] but if this condition is not true and n[i]<n[bpos] then the while loop has the capacity to continue the loop till i<10 and it searches the next element for finding this condition and search to whole array run till the condition becomes true;at which location this condition matches. It changed the I in bpos as bpos=I and print the location of number.

Same as in the second condition we kept that n[i]<n[spos] it will search for the smallest no. in array & print it.

We know that if we take an integer according to the structure of the program, we will have to put 10 no's in array & here we can test this program by making the entry values.

(i) if we put different no's in array as 1,5,11,10,6,15,20,21,2,8

We will find the desired no. as bpos we will get 21 and spos as 1 and the location will say 7 & 0.

(ii) If we put some repeatable no. as input as 1,5,11,10,11,15,20,21,2,8

We will find the desired no. as bpos and spos as its location I 7 & 0.

(iii) If we put the biggest no. two or three times as input or we put the smallest no. 2 or 3 times as input as 1,5,1,11,10,11,15,12,21,2 then it will give the first no. as it will traverse till it finds the biggest & smallest at first time according to this the output is biggest element 7 and smallest element 0.

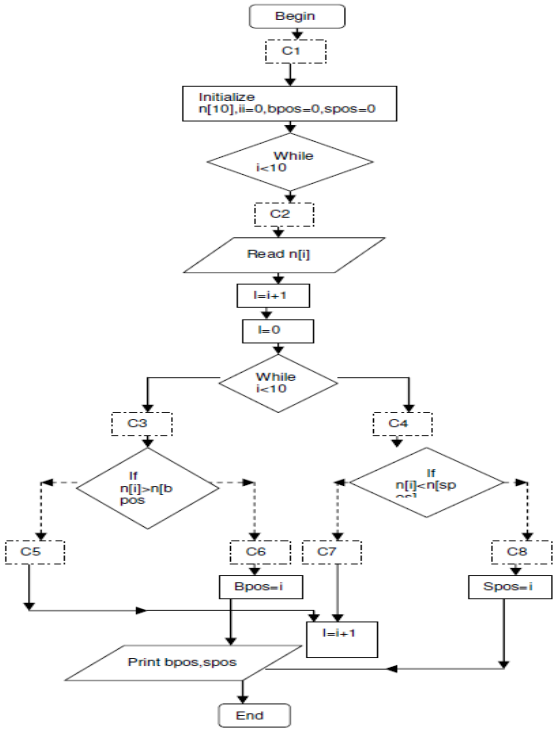
There is a class of common programming error that can not be discovered in this way. The problem is that a program may contain paths from the entry to the exit; (in its flowchart) which need not be traversed in order to have every element executed at least once.

**Problem in the above criteria**: After testing the problem arises that this criteria was impractical and leaves the undetected errors during program testing.

**Criteria decided:** "Every branch of the flowchart of program will be executed at least once during the program testing"[13].

**Problem in the above criteria :**A class of common programming error could not find in this criteria (way) because a program contains the path from entry to exit and need not to be traversed in order to have every statement will executed at least once ,so this is also condemned.

## Criteria decided:




"Every control path in the program is traversed at least once during the test as now we have tested the program for a set of test cases".[13] The used test cases and resulting counter values that counts and onwards becomes o by inspection we found that here we have not changed the variable in array and next.

**Problem in the above criteria:** It is impractical because every program contains the loops, and a program with a loop contains at least as many different control paths as the number of times the loop can be iterated, which is prohibitively large in many cases.

## IX. REQUIREMENT OF ADOPTING ADVANCED DEGREE AUTOMATION TECHNIQUE

Advanced level techniques have the features of executing thousands (of billions) of tests without human inference.

**Functional equivalence testing:** in this type of testing we have generated random input data and have used to compare the behaviors of the function under test with reference program.

On between performance and functionality testing: performance testing is a high-margin business, needing highly skilled man-power, while projects themselves are sporadic; functionality testing is low margin requires a lot of man-power but can help sustain an organization.

**Random less advanced stage testing:** In this we have run the test in random order, for checking whether the program continues to pass them over time.

And found that this style of testing highlights problems like graded memory corruption.

**Path testing**

Advanced testing simply covers a much higher proportion of tests in a well understood set it reaches the risks(input error, programming error, insufficient coding error, improper modeled flowchart errors) these are harder to troubleshoot, especially if the failure happens.

## X. PREDICTIONS ABOUT THE FUTURE OF SOFTWARE TESTING

Techniques for making testing more effective and efficient the change in s/w testing to is to develop in parallel.

For the development of the quality, more emphasis must be placed on improving the analysis and design phases.

The design and analysis should be based on the natural language, models and diagramming techniques (such DFDs, logical data structure, logical data models and process data diagrams and some structured language techniques (such as use cases & UML)[9]

The future will be enable about the production of quality applications and analysis &design technologies will more structured and more automated with static testing (reviews, inspections & walk throughs will be 'robotized ' & built as part of the analysis and design requirements capture tools.

Testing would be enabling as:

Executable Specifications[4]

Auto generated code and systems configurations

Model based test generation

Model based bug prevention

System simulation

Fault tolerance developing into self testing, self monitoring, and self healing software.Performance optimization "designed in "and "n the fly".

Testing and Quality assurance will become more important and add more value & therefore it will mitigate the risks and increasing the benefits. Raising the quality of software without recourse to elaborate techniques or complex methodologies is still a needed and practical solution.

## XI. CONCLUSION

Testing business has its unique challenges. Recently testing was an unpopular career and as a result current test resource base is poor but now in recent years, software test as a quality assurance and risk management activity gains a lot more attention and appreciation. But the principal challenge of shaping and managing the profession of software testing-its standards, skills, techniques and practices-belongs to us.

## XII. FUTURE STRATEGY

Future of software testing will change the areas:-

- Accurate information provision (in the language of the recipient)
- Building quality in, rather than testing it out,prevention rather detection
- Ensuring that quality is considered and not seen as a burden ,alongside time and cost.
- To avoid the situation of software accidents and problems e-Japan strategy by all citizens it japan; It promotes the researcher to study on software terminology, description and representation and basic knowledge.

Published in Eurostar Newsletter August 2008 It says *"The era of the independent tester is over". to assure the future of testing : creative work should be done".*

- *Develop new ways to test better and faster.*
- *Select the best and most appropriate techniques, tools and practices for each job from a well-stocked and constantly growing bag of tricks.*
- *Decide how we can best contribute our skills, as project strategies and development methods inevitably grow and change.*
- *Tester should take courage to work side by side with developer colleagues*

## REFERENCES

[1] "Technical trends and challenges of software testing" Toshiaki kurokawa/masato shinagawa Affiliated fellow, 2008

[2] Software testing: main trends in test and measurement (Wednesday,,January 14,09)by "Sergey Lesnikov"

[3] "The future of software testing-test management", "Computer world UK",August 21,2008.

[4] "The future of software testing is ours to make!", @2008 "Fiona Charles" published in Euro star newsletter august 2008

[5] Vibrant trends in software testing (News and Events, amity soft), 31st July 2006.

[6]"Software testing Techniques-Technology Maturation and Research Strategy" ,Lu Luo (Institute for software Research International Carnegie Mellon University, Pittsburgh, "PA 15232,USA.

[7] The top five software testing problem and how to avoid them", Lars Mats, Teleological Technologies Malmoa 13-EDN,2/1/2007.

[8] Inefficiency and Ineffectiveness of software testing :A key problem in software Engineering ,Cem Kaner,Ph.D..J.D.

[9]" Future of Software Testing",AppLabs,7 July 2008.

[10]" Innovative Technology for Computer Professionals", Computer IEEE,Computer Society,2008

[11] "Testing Large Software with automation software Evaluation Systems",C.V. Ramamoorthy and S.F.Ho,Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory,University of California,Berkeley, California,1978.

[12] "Toward A Theory of Testing Data Selection Criteria",John B. Goodenough Softech,Inc Waltham,Massachusetts,Susan L. Gerhart,Duke University,Durham,North Carolina,1978-79

[13]" Error Detection through Program testing",J.C.Haung,Department of Computer Science,University of Houston,Houston,Texas,1975

[14] "Modern Book on Software Testing(Tools and technique),Rajan Nagia,2008

[15] "Randy Rice", Residenza Di Ripetta-Via Di Ripetta ,231 Rome(Itlay)