International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Use Of V & V Strategy as Project Management

Swati Gupta[1]
Arshpreet Kaur[2]
Kumar Shashvat[3],
Mrs. Ramandeep Sandhu[4]

Abstract

Software Testing is a tedious job and accomplishing software verification and validation (V&V) activities is not a simple task. It involves a great number of techniques to choose and there is no sufficient organized information to support the selection regarding the V&V technique to be used in the Re-engineering process. This paper describes a method concerned with the definition of an approach to plan verification and validation processes so that it could be implemented in the Reverse engineering process

Keywords- Testing Spectrum, Testing objectives, Testing Principle, Testing Techniques, Verification, Validation, Software Re-engineering

1 M. Tech- CSE- CGC Technical Campus, Jhanjeri, Mohali
2 M. Tech- CSE- CGC Technical Campus, Jhanjeri, Mohali
3 M. Tech- CSE- CGC Technical Campus, Jhanjeri, Mohali
4 Associate Professor- CSE - CGC Technical Campus, Jhanjeri, Mohali

Introduction

Software testing is the process of executing a program or system with the intent of finding errors. 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. The aim of testing is to identify all defects in a software product, However for most practical systems even after satisfactorily carrying out the testing phase it is not possible to guarantee that the software is error free. This is because of the fact that input data domain f most software product is large. Even with the practical limitation of the testing process we should not underestimate the importance of testing as it expose many defects existing in software product.

Testing a program consist of subjecting the program to a set of inputs and observing if the program behaves as expected. Following are some commonly used terms associated with testing.

- **Failure:-** Failure can be defined as the deviation of the delivered service from the function the program was intended for the originating cause of the failure is said as fault.
  Fault $\rightarrow$ Error $\rightarrow$ Failure

- **Test Case:-** A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. Test case could be represented as a triplet [I, S, O] where I is a combination of inputs, S is executing function & O is expected outputs.
- **Test Suite:-** Test suites are the collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviors. A test suite for a primarily testing subroutine consist of a list of numbers and their primarily (prime or composite), along with a testing subroutine

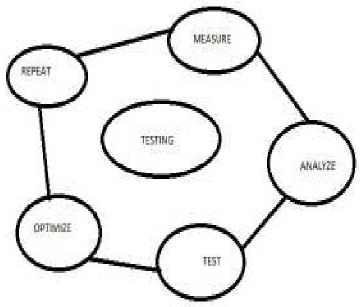


Fig 1: Various operations in testing process.

Objective

The objective of testing is to find problems and fix them to improve quality. Software testing typically represents 40% of a software development budget. There are four main objectives of testing:

- Affirmation: It shows that, system developed could be used for unification with acceptable risk. It represents functions under special conditions and depicts that products are ready for amalgamation or use.
- Perception: It identifies defects, errors and deficiencies. It also dictates system capabilities and constraints standard of components, work products and the system.
- Interception: It provides knowledge to prevent or minimize the number of delusion to clarify system specifications and performance. Discover ways to avoid risk and problems in the future.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

- Upgrade Quality: By doing constructive testing, we can reduce errors and hence improve the quality of software.
-

III. TESTING PRINCIPLE

Before applying methods to design productive test cases, a software engineer must implement the various testing services and comprehend the basic principles that guide software testing.

1. All tests should be verifiable to customer necessity. As we know, the objective of software testing is to uncover errors. It follows that the most severe imperfection (from the customer's point of view) are those that cause the program to fail to meet its needs.
2. Tests should be planned long before testing begins. Test planning can start as soon as the requirements model is intact .Comprehensive definition of test cases can start as soon as the design model has been solidified. Therefore, all tests can be planned and designed before any code has been generated.
3. The Pareto principle relate to software testing. which implies that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components. The difficulty is to isolate these suspect components and to properly test them.
4. Testing should begin "in the small" and progress toward testing "in the large." The first tests planned and executed generally target on discrete components. As testing progresses, focus shifts in an attempt to perceive errors in integrated collection of components and ultimately in the entire system
5. Exhaustive testing is not possible. The number of path permutations for even a moderately sized program is exceptionally huge. It is possible, adequately cover program logic and to guarantee that all conditions in the component-level design have been exercised. To be most effective, testing should be conducted by an independent third party. By most effective, we mean testing that has the highest probability of finding bugs.

IV . TESTING SPECTRUM

Testing is included in every stage of software life cycle, but the testing done at each level of software development is different in nature and has different objectives.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

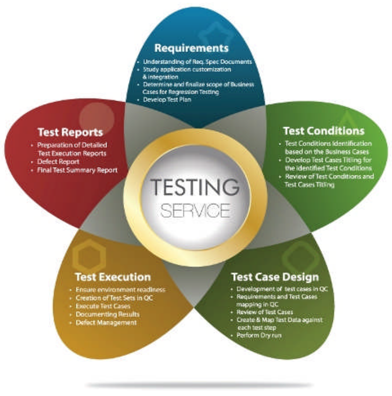

Requirements
Understanding of Req. Spec Documents
Study application customization & integration
Determine and finalize scope of Business Cases for Regression Testing
Develop Test Plan
Test Reports
Preparation of Detailed Test Execution Reports
Defect Report
Final Test Summary Report
Test Conditions
Test Conditions Identification based on the Business Cases
Develop Test Cases Titling for the identified Test Conditions
Review of Test Conditions and Test Cases Titling
TESTING
SERVICE
Test Execution
Ensure environment readiness
Creation of Test Sets in QC
Execute Test Cases
Documenting Results
Defect Management
Test Case Design
Development of test cases in QC
Requirements and Test Cases mapping in QC
Review of Test Cases
Create & Map Test Data against each test step
Perform Dry run

Fig 2: Testing Services

- **Functional Testing:** Functional testing is a quality assurance (QA) activity and a category of black box testing that support its test cases on the detailing of the software component under test.

- **Unit Testing:** Unit Testing is done at the moderate altitude. It tests the fundamental unit of software, which is the insignificant testable piece of software, and is often called “unit”, “module”, or “component” interchangeably.

- **Exploratory Testing:** Exploratory testing is an outlook to software testing that is concisely characterize as simultaneous learning, test design and test execution. Exploratory testing is always been performed by professional testers.

- **Ad-Hoc Testing:** Ad hoc testing is an informal and extemporization approach to evaluate the vitality of a product. An ad hoc test is usually conducted once unless a defect is found. Exploratory testing is often referred as refinement of the ad hoc model.

- **Integration Testing:** Integration Testing is accomplish when two or more tested units are incorporated into a larger structure. The test is often done on both the interfaces between the components and the massive structure being constructed, if its quality property cannot be assessed from its components.

- **User Acceptance Testing:** Acceptance testing is a test conducted to diagnose if the demand of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests.

- **Usability testing:** Usability testing is an approach used in user-centered interaction design to gauge a product by testing it on users. Usability testing focuses on computing a human-made product's potential to meet its intended justification.

- **Alpha Testing:** Alpha testing is counterfeit or actual operational testing by potential users/customers or an autonomous test team at the developers' site. Alpha testing is often utilized for off-the-shelf software as a form of intramural acceptance testing, before the software goes to beta testing.

- **Beta Testing:** In software development, a beta testing is the secondary phase of software testing in which a appraisal of the intended audience endeavor the product out. (Beta is the second letter of the Greek alphabet.)

- **Compliance Testing:** It is a type of non-operational software testing. It determines, whether we are executing and gathering the defined benchmark. Its basically an scrutiny of a system carried out against a known barometer.

- **Functional Testing:** Functional testing is a quality assurance (QA) process and a type of black box testing that essence its test cases on the identification of the software component under test.

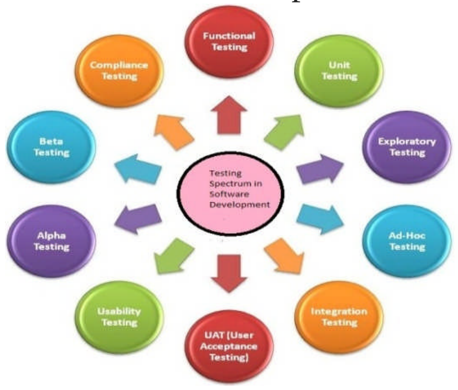


Fig 3: Testing Spectrum

V. TESTING TECHNIQUE

There are various types of testing techniques that have been invented. Each testing technique serves a different purpose for testing different artifacts like designing, coding and planning software requirement specification.

White Box Testing: (WBT) is also known as Code-Based Testing or Structural Testing. White box testing is the software testing method in which internal structure is being known to tester who is going to test the software. It is a Testing based on an analysis of the internal structure of the component or system.

Black Box Testing: Black Box Testing is testing without knowledge of the internal workings of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs.

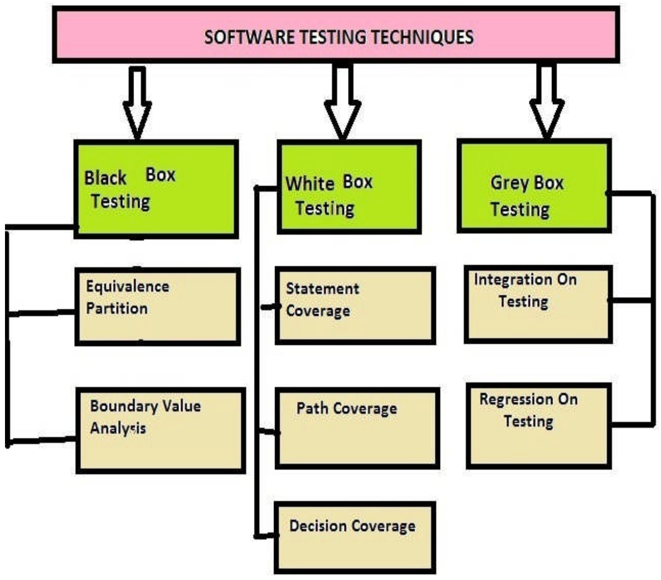


Fig 4: Software Testing Techniques

Grey Box Testing: Grey Box testing is testing technique performed with limited information about the internal functionality of the system. Grey Box testers have access to the detailed design documents along with information about requirements. Grey Box tests are generated based on the state-based models, UML Diagrams or architecture diagrams of the target system.

Black Box Testing
White Box Testing
Grey Box Testing

Fig 5: Testing process.

VI. Software Testing life cycle

Software Testing Life Cycle is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product..

Requirement Analysis:

Requirement Analysis is the very first step in Software Testing Life Cycle (STLC). In this step Quality Assurance (QA) team understands the requirement in terms of what we will testing & figure out the testable requirements. If any conflict, missing or not understood any requirement, then QA team follow up with the various stakeholders like Business Analyst, System Architecture, Client, Technical Manager/Lead etc to better understand the detail knowledge of requirement.

From very first step QA involved in the where STLC which helps to prevent the introducing defects into Software under test. The requirements can be either Functional or Non-Functional like Performance, Security testing. Also requirement and Automation feasibility of the project can be done in this stage

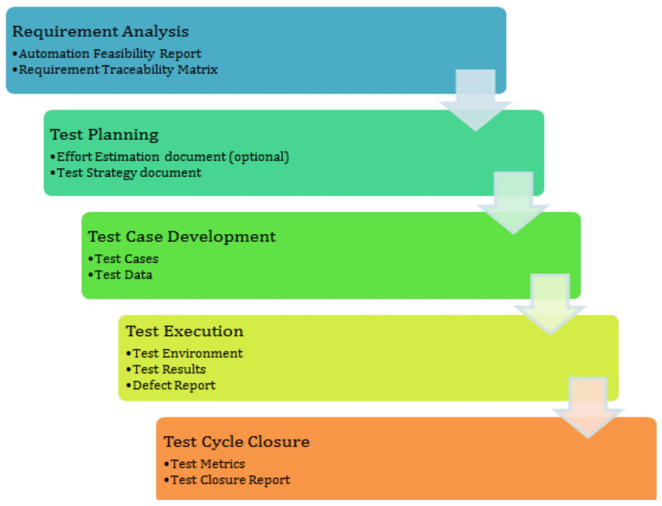


Fig 6: Software Testing Life Cycle Process

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Test Planning (Product Definition Phase):

The test plan phase mainly signifies preparation of a test plan. A test plan is a high level planningdocument derived from the project plan (if one exists) and details the future course of testing.

Test Case Development:

The test case development activity is started once the test planning activity is finished. This is the phase of STLC where testing team write down the detailed test cases. Along with test cases testing team also prepare the test data if any required for testing. Once the test cases are ready then these test cases are reviewed by peer members or QA lead.

Also the Requirement Traceability Matrix (RTM) is prepared. The Requirement Traceability Matrix is an industry-accepted format for tracking requirements where each test case is mapped with the requirement. Using this RTM we can track backward & forward traceability.

Test Execution (Unit / Functional Testing Phase)

By this time. the development team would have been completed creation of the work products. Of Course- the work product would still contain bugs. So, in the execution phase - developers would carry out unittesting with testers help, if required. Testers would execute the test plans. Automatic testing Scripts wouldbe completed. Stress and performance Testing would be executed. White box testing, code reviews, etc.would be conducted. As and when bugs are found - reporting would be done

Test Cycle Closure (Re-Testing Phase)

By this time, minimum one test cycle (one round of test execution) would have been completed and bugswould have been reported. Once the development team fixes the bugs, then a second round of testingbegins. This testing could be mere correction verification testing -- that is checking only that part of thecode that has been corrected. It could also be Regression Testing - where the entire work product istested to verify that correction to the code has not affected other parts of the code. Hence this process of :

Testing --> Bug reporting --> Bug fixing (and enhancements) --> Retesting

is carried out as planned.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

The Proposed Work

Implementation of Verification and Validation in Reverse Engineering using Software Testing Life Cycle Process

Verification: Verification is the process of checking that the software meets the specification. It includes all the activities associated with the producing high quality software. It establishes the truth of correspondence between a work product and its specification.

Validation: Validation establishes the fitness of a software product for its operational mission. It is concerned with checking that the system will meet the customer's actual needs. It is relegated to just the beginning and ending of the project.

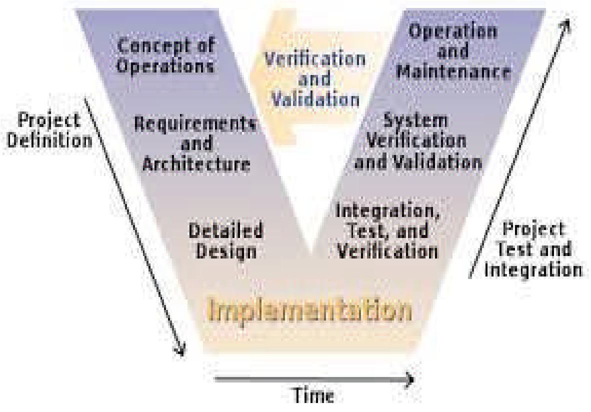
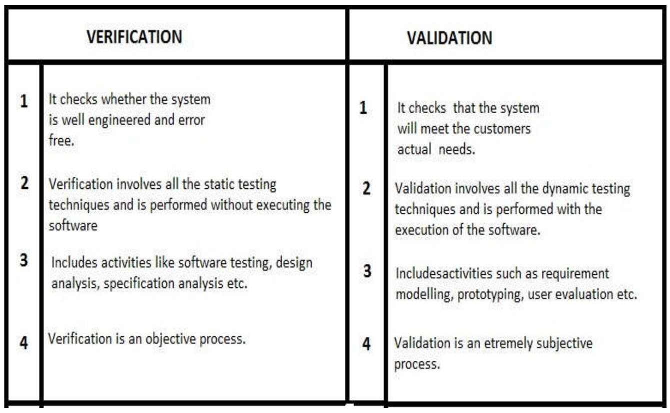

Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Fig 7: Verification and Validation implementation



	VERIFICATION		VALIDATION
1	It checks whether the system is well engineered and error free.	1	It checks that the system will meet the customers actual needs.
2	Verification involves all the static testing techniques and is performed without executing the software	2	Validation involves all the dynamic testing techniques and is performed with the execution of the software.
3	Includes activities like software testing, design analysis, specification analysis etc.	3	Includesactivities such as requirement modelling, prototyping, user evaluation etc.
4	Verification is an objective process.	4	Validation is an etremely subjective process.

Fig 8: Differences between Verification and Validation

Reverse engineering: Reverse engineering is the examination, analysis and alteration of an existing software system to reconstitute it in a new form, and the subsequent implementation of the new form. The process typically encompasses a combination of other processes such as reverse engineering, redocumentation, restructuring, translation, and forward engineering. The goal is to understand the existing software (specification, design, implementation) and then to re-implement it to improve the system's functionality, performance or implementation. The objective is to maintain the existing functionality and prepare for functionality to be added later.

However, since our approach of supporting Verification and Validation in Reverse engineering activities will be supported by experimental knowledge, some new challenges to attend these traditional steps must be observed:

• **Knowledge Creation and Capture**: Knowledge needs to be created or imported form the existed software. An important challenge regarding creation and capture of experimental knowledge is how to generate knowledge and evidences through many developed product. Another important challenge is how to support an organization to create experimental knowledge, that is, to perform its own software re-engineering experiments, and aggregate its results into organizational memory.

• **Knowledge Retrieval and Access**: These steps satisfy the searches and queries for the process of re-engineering. An important challenge is how to describe experimental knowledge items to support retrieval and future accesses. Its proposed schema describes knowledge regarding testing techniques. However, extensions should be provided to support the representation of different types of verification and validation techniques.

• **Knowledge Use:** The developer will recall knowledge item, and will process them for further use. A challenge such as integration with the user's task plays a crucial role for the effective use of experimental knowledge. We have to associate such knowledge to the software process or even embed it into CASE tools to support the execution of software testing activities.

• **Knowledge Maintenance and Evolution Facilities**: A challenge of these steps is to define how organizational memory should aggregate results of new experimental studies and, also, how to evaluate its repository to decide what knowledge item is obsolete or which one had never been used in the process of Software Testing.

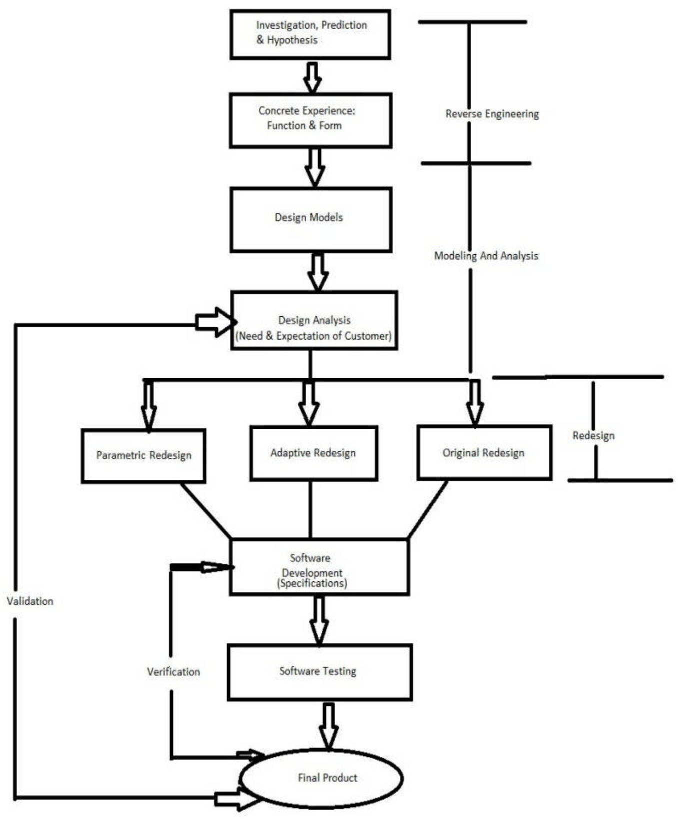


Fig 9: Implementaion of Verification & Validation in Reverse Engineering

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Conclusion

V&V methodology targets testing which is beneficial for managing project in all factors. Reverse engineering is one of the focusing world- linked method of software world which is towards producing high quality work.

References

[1]http://www.ijsce.org/attachments/File/v2i3/C0761062312.pdf

[2]https://sites.google.com/site/itsallabouttesting/Home/fundamentals-of-testin

[3]https://www.academia.edu/1743243/Software_Testing_Life_Cycle

[4]http://www.cis.upenn.edu/~lee/05cis700/papers/Ber03.pdf

[5]http://www.tutorialspoint.com/software_testing_dictionary/grey_box_testing.html
[6]http://www.softwaretestinghelp.com/what-is-verification-and-validation/