Need for VV&A (Verification, Validation and Accreditation) In Software Organizations

Mohammed Rafiq

Research Scholar (Computer Science), CMJ University, Shillong, Meghalaya

Abstract— the software testing is the process, which involves many non-technical and technical areas such as specification, process and management issues, maintenance, implementation and design in the software engineering. The characteristics such as execution efficiency, usability, accuracy, portability, maintainability and reusability are most important in the development phase of software engineering. Software testing is one of the processes of analyzing the software to find the differences between required and existing conditions and also to evaluate the features of the software. Verification, Validation and Accreditation (VV&A) plays main role in software testing process. Verification, Validation and Accreditation helps to develop and assure quality in modeling and simulation which are integrated in the program.

Index Terms— Software Testing Verification, Validation and Accreditation

----------♦----------

1. INTRODUCTION TO VERIFICATION, VALIDATION AND ACCREDITATION

Verification is the process of evaluating the work products of development phase in order to determine about whether they meet the requirements that are needed for that phase. Verification is to ensure that the product is built with specific design specifications and requirements. In other words, verification is to ensure that the products have capabilities to meet the specific requirements.

Validation is the process of evaluating the software at the end or during the development process in order to determine about whether the software satisfies the required or specified business requirements (Sommerville, 2000). Validation is the process to ensure that the products have capabilities to meet the needs of the user and also the specifications were correctly used. In other words, validation is the process which demonstrates that the product may have capability to fulfill its intended use in its intended environment.

Accreditation is the process that helps in the certification of competency, authority or credibility is presented (Rushby and John, 1993). Generally, accreditation ensures that their certification practices are competent to test and certify third parties.

a) VERIFICATION:

Verification process will represent the static testing techniques. Verification process may ensure that the software documents will comply within the organizations standards (Kaner, Bach, and Pettichord, 2002). The Verification Strategies are: Requirements Review; Code Inspections; Design Review; and Code Walkthrough. In addition to these, there are some important verification techniques and they are: Feasibility reviews; Technical Reviews; Inspections; Informal reviews; Formal reviews; and Static Code Analysis.

The following table illustrates the explanation of verification strategies.

Verification Strategy	Explanation	Deliverable
Requirements Review	The study and discussions of the computer system requirements to ensure they meet stated user needs and are feasible.	Reviewed statement of requirements.
Design Review	The study and discussion of the computer system design to ensure it will support the system requirements.	System Design Document, Hardware Design Document.
Code Walkthrough	Informal analysis of the program source code to find defects and verify coding techniques.	Software ready for initial testing by the developer.
Code Inspection	Formal analysis of the program source code to find defects as defined by meeting system design specification.	Software ready for testing by the testing team.

Table: Verification Strategy

Source: Powell, P. B (1986): "Planning for Software Validation, Verification, and Testing." In Software Validation, Verification, Testing and Documentation, S. J. Andriole, ed. Princeton, N. J.: Petrocelli, 1986 3-77.

The two verification criteria are: the program that run on the particular computer and also it must satisfy the particular specification; the specification that given in the domain properties must satisfies the needed requirements.

b) VALIDATION:

Validation process will represent the dynamic testing techniques. Validation process may ensure that the software will operates according to what they have planned in the requirements phase and by executing it with predefined test cases and also by measuring the output with expected results (Craig and Jaskiel, 2002). Validation process is concerned with evaluating the component or system and software in order to determine how it meets the end user requirements. The Validation Strategies are: Unit Testing; Integration Testing; Performance Testing; System Testing; Alpha Testing; Installation Testing; User Acceptance Testing (UAT); and Beta Testing.

The following table illustrates the explanation of validation strategies.

Validation Strategy	Explanation	Deliverable
Alpha Testing	Testing of the whole computer system before rolling out to the UAT.	Stable application.
User Acceptance Testing (UAT)	Testing of computer system to make sure it will work in the system regardless of what the system requirements indicate.	Tested and accepted system based on the user needs.
Installation Testing	Testing of the Computer System during the Installation at the user place.	Successfully installed application.
Beta Testing	Testing of the application after the installation at the client place.	Successfully installed and running application.
Unit Testing	Testing of single program, modules, or unit of code.	Software unit ready for testing with other system component.
Integration Testing	Testing of related programs, modules, or units of code.	Portions of the system ready for testing with other portions of the system.
System Testing	Testing of entire computer system. This kind of testing can include functional and structural testing.	Tested computer system, based on what was specified to be developed.
Performance Testing	Testing of the application for the performance at stipulated times and stipulated number of users.	Stable application performance.

Table: Validation Strategy

Source: Powell, P. B (1986): "Planning for Software Validation, Verification, and Testing." In Software Validation, Verification, Testing and Documentation, S. J. Andriole, ed. Princeton, N. J.: Petrocelli, 1986 3-77.

The two validation criteria are: it must understand and discover all the important requirements; and it must understand and discover all the relevant domain properties. The following table illustrates the differences between verification and validation.

Verification	Validation
Verification represents static testing techniques.	Validation represents dynamic testing techniques.
Verification ensures that the software documents comply with the organizations standards, it is static analysis technique.	Validation ensures that the software operates as planned in the requirements phase by executing it, running predefined test cases and measuring the output with expected results.
Verification answers the question "Is the Software build according to the specifications".	Validation answers the question "Did we build the software fit for purpose and does it provides the solution to the problem".

Table: Differences between verification and validation

Source: Hetzel, W. (1984): The Complete Guide to Software Testing, QED Information Sciences, Wellesley, MA.

2. VERIFICATION, VALIDATION AND ACCREDITATION (VV&A) PRINCIPLES:

Verification and Validation is one of the processes of checking whether the software system has capability to meet the particular specifications and also to fulfill its intended purpose. Verification and Validation process may also referred to as the software quality control (Pressman, 2001). The terms Validation and Verification are most frequently used in software testing and the meaning of those terms are mostly debatable and vague. Verification, Validation and Accreditation principle can be defined as tenet or fundamental doctrine; general, primary or fundamental law or the truth from which others are derived; and professed or accepted rule of conduct or action. Principles are the most important one in order to

understand the foundations of Verification, Validation and Accreditation.

The following table illustrates the basic principles of Verification, Validation and Accreditation.

1	V&V must be conducted throughout the entire M&S life cycle.
2	The outcome of VV&A should not be considered as a binary variable where the model or simulation is absolutely correct or absolutely incorrect.
3	A simulation model is built with respect to the M&S objectives and its credibility is judged with respect to those objectives.
4	V&V requires independence to prevent developer's bias
5	VV&A is difficult and requires creativity and insight.
6	Credibility can be claimed only for the prescribed conditions for which the model or simulation is verified, validated and accredited.
7	Complete simulation model testing is not possible.
8	VV&A must be planned and documented.
9	Type I, II and III errors must be prevented
10	Errors should be detected as early as possible in the M&S life cycle.
11	Multiple response problems must be recognized and resolved properly.
12	Successfully testing each sub model (module) does not imply overall model credibility.
13	Double validation problem must be recognized and resolved properly.
14	Simulation model validity does not guarantee the credibility and acceptability of simulation results.
15	A well-formulated problem is essential to the acceptability and accreditation of M&S results.

Table: Principles of Verification, Validation and Accreditation.

Source: Adrion, W.R., Branstad, M.A., and Cherniavsky, J.C. (1982): "Validation, verification and testing of computer software", ACM Computing Surveys 14, 159-192.

3. NEED FOR VERIFICATION VALIDATION AND ACCREDITATION IN SOFTWARE ORGANIZATIONS

Verification and Validation processes should be bring throughout the software evolution process. Verification and Validation processes helps to identify the typical products of the software process along with the objectives of Verification and Validation. Verification and Validation processes may consists of many tools and techniques and they may also be used with the combination of one another. Verification and Validation (V&V) is the process of ensuring that the software design meets its requirements (Adrion, Branstad, and Cherniavsky, 1982). Verification assures that the products are properly reflecting the specified requirements and ensures that "programmer built it right". Validation assures that the product that provided was fulfilling its intended use and also ensures that "programmer built the right thing".

The following figure illustrates the process of verification and validation.

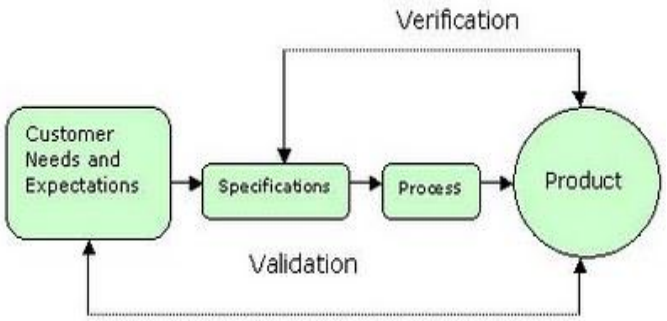

Table: Process of Verification and Validation

Source: Banks, J. (1989), "Testing, understanding and validating complex simulation models," in: Proceedings of the 1989 Winter Simulation Conference.

Verification, Validation and Accreditation is most important in software testing and this is because: it have capability to provide an understanding of the capabilities, assumptions and limitations of the models and simulations; and it assures how trustworthy the results of the process; and also it promotes reuse of process by allowing others to understand how the process are used in there (Beizer, 1984).

4. SOFTWARE VERIFICATION AND VALIDATION:

Software verification provides the evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. Verification of the software is for completeness, correctness and consistency of the software and also for supporting documentation, which have been developed and also to provide the support for the subsequent conclusion that the software is validated (Miller and Howden, 1981). Verification also includes some other activities like dynamic and static analyses, walkthroughs,

document and code inspections and other techniques. Software testing is also one of the verification activities that are intended to assure that the software development meets its requirements.

Software validation provides the evidence that the software specifications assure to the user needs and intended uses and also the requirements that are implemented through the software are consistently fulfilled. Generally, software validation activities will occur either at the end or during the software development life cycle in order to ensure that all the specified requirements have been fulfilled (Sargent, 2003).

Verification process in software testing would takes place before the validation process. Validation activities (user feedback, usability testing, etc.) are so harder to execute, document and define properly than the verification testing (Andriole and Stephen, 1986). Verification process evaluates the plans, requirements, code, specifications and documents. On the other hand, validation process evaluates the product itself. Software validation involves the process of execution of the tests that are designed in order to cover the specified system requirements.

5. CONCLUSION

It is concluded that software testing is one of the techniques for detecting the errors in the software development phase. The main aim of the software testing is assure the quality of the software. Software testing is to detect the correctness of the software and the software should be tested more efficiently in order to have better software. It is concluded that Verification and Validation (V&V) is the process that helps for checking the software system whether it meets its specifications and also it fulfills the intended purpose. Verification and Validation is the part of software testing process in the software development phase. It is concluded that is most Verification and Validation (V&V) is most important testing process in the software development process. Verification, Validation and Accreditation (VV&A) plays an important role in software testing.

REFERENCES

1. Adrion, W.R., Branstad, M.A., and Cherniavsky, J.C. (1982), "Validation, verification and testing of computer software", ACM Computing Surveys 14, 159-192.

2. Sommerville I, (2000): Software Engineering, Addison-Wesley, 6th edition, August 2000.

3. Pressman, R. (2001), Software Engineering: A Practitioner's Approach, Boston, McGraw Hill.

4. Kaner C, J. Bach, and B. Pettichord, (2002): Lessons Learned in Software Testing: John Wiley & Sons.

5. Craig R D and S. P. Jaskiel, (2002): Systematic Software Testing. Norwood, MA: Artech House Publishers, 2002

6. Beizer, B (1984): Software System Testing and Quality Assurance. New York: Van Nostrand, 1984.

7. E. Miller and W. E. Howden (1981): Tutorial: Software Testing and Validation Techniques, 2nd Ed, Los Alamitos, Calif.: IEEE Computer Society Press.

8. Sargent, R. G. (2003): "Verification and Validation of Simualtion Models" 2003 Winter Simulation Conference. Chick, S., P.J.Sanchez, D. Ferrin and D. J. Morrice, eds. pp. 37-48.

9. Andriole, Stephen J., (1986): Software Validation, Verification, Testing, and Documentation, Princeton, NJ: Petrocelli Books, 1986.

10. Rushby, John (1993): "Formal Methods and the Certification of Critical Systems," SRI-CSL Technical Report, November 1993.