

EVALUATION OF THE DIFFERENCE BETWEEN VERIFICATION AND VALIDATION OF SOFTWARE AND ANALYZING THE SIGNIFICANCE AMONG BOTH

YAZEED GHADI, MOHAMMAD SH. DAUD, FATEN KHARBAT AND TARIK ELAMSY

Faculty of Engineering

Al Ain University of Science and Technology

P.O.Box: 112612, Mohammed Bin Zayed City, Abu Dhabi, United Arab Emirates
{ yazeed.Ghadi; mohammad.daoud; faten.kharbat; tarik.elamsy }@aau.ac.ae

Received April 2019; accepted July 2019

ABSTRACT. *Software validation and verification is an important process within the software development life cycle used in the development of software systems. Software validation and verification are closely linked to the software system structure analysis and design where verification is an important part of identifying different purpose and significance of their software products. On the other hand, the major role of validation is to improve the efficiency of the software system. Both verification and validation are the essential parts for software development and used to improve the performance of their system program. Moreover, both processes are maintaining a proper balance in an automated system. This paper aims at highlighting the differences between the V-models and analyzing their significance.*

Keywords: Software development life cycle, Software validation process, Verification process, Software testing process, Debugging process

1. Introduction. Development of information system is helpful to a huge number of software engineers as well as exploited by the engineers of instrumentation and control in process industries. This has subsequently improved and ensured satisfactory levels of the performance of industries and plants. The systems that are associated with the use of computers or related technology were installed initially for functions that were necessary for industrial operations. However, they were not at all related to the safety of the systems. Introduction of the software-based systems approximately 2 decades ago, has led to examining well-documented challenges and difficulties across various industrial sectors. Software validation and verification are the two most essential parts of the Software Development Life Cycle (SDLC) which are mainly used for evaluating or executing the program. In general, verification is used to identify each and every issue related to the programs [1].

On the other hand, validation is used to check whether the software system meets every specification or not [2]. Verification and validation are known as the most commonly used technique through which every IT industry could be able to ensure their product performance and specifications. Every device or system is developed and maintained using proper practices and all the risks related to the use of products are generally reduced through reasonable verification. In order to verify any software, it is essential that they also have to focus on the specification related to the program. In the case of software verification, it does not suffer from physical changes. The process of software verification and modification is dependent on user requirements. Software verification or modification requires to depend upon the user requirements [3]. However, during the modification or validation process, the requirements are critically verified and tested, through which all

the developers could be able to upgrade their software system. In order to run a proper validation process, all the developers have to focus on the different specifications of their system and users requirements [4]. In a software development life cycle, validation of software undertakes all the responsibilities to check the entire specification as well as the purpose of that software system. In addition, in case of verification or modification, the developers need to involve a debugging process through which they could be able to verify all the specification relates to their identified issues. However, the main objectives of the validation and verification process can be identified as, testing or checking items including code and conformance through observing the results. The process of debugging implies when the aim of that program goes wrong and it requires a proper modification [5]. The scope of this process can be specified as, this validation or verification process mainly used to check whether the system meets customer's requirements.

This paper focuses on the understanding of the process to validate software. In this respect, aim, objectives and various aspects of software validation are discussed. Along with this, the deductive research approach is taken into consideration to have a specific result over the concerned scenario. The secondary research is performed based on the V model of software development and verification.

The aim of this research paper is to evaluate the differences between software verification and validation in accordance with their significant contribution to software quality. The paper also highlights on analyzing the more important one between these two types of testing processes. The objectives of this research paper include the identification of the principles of software verification and validation and analyzing their needs. The paper also examines the need for validation and verification of software and identifies the basic differences between both.

The information obtained from software validation and verification allows identifying the requirement specifications of the system focusing on the fact that whether they are concise, complete, consistent, correct and unambiguous. By the end of this paper, a piece of clear information regarding the differences between software validation and verification is expected while taking due account of the prevalent regulatory procedures. This paper primarily intends in assisting users, suppliers, and regulators in developing, installing, accepting and licensing of systems that are computer-based contributing to their safety. Justification of the approaches of verification and validation of the software will enhance the quality plan of the development process.

In Section 2, the fundamental principles, differences and best practices of software validation and verification are reviewed. In Section 3, we present our research methods. Results and analysis of the research are presented in Sections 4 and 5 respectively followed by a conclusion.

2. Related Work.

2.1. Principles of software validation and verification. In order to run a proper software system, it is essential to verify all the specifications and requirements of that system. It is also essential to check the validity of that system so that all the developers could be able to undertake preparation for further modification. Software verification mainly focuses on various objectives of the software development process, through which they could be able to provide evidence that, the cycle has met all the requirements [6]. Software verification process needs to be done properly so that the entire system could meet its targets. Moreover, software verification looks after completeness, correctness, and consistency of the software and its supporting systems. Verification is mainly used to check whether the system requires any kind of modification. The verification process includes code and document inspections, dynamic and static analysis and so on which is totally based on the different specification of that program [7].

Software validation is one of the most important parts of SDLC that is used for a completed device. This process is mainly used as evidence, through which they could be able to provide a verified description of their different specifications of the software system. The software validation process is involved at the end of the software development life cycle so that it could ensure all the specifications meet the requirements [7]. In case of a hardware system, a software validation system was used to verify whether all the specifications are connected properly with the hardware system. However, the validation of a software system is dependent upon the software analyzing, testing, verification, and another testing process at the stage of the software development life cycle.

The process of software development is too complex for developers because they cannot test each individual program forever. As a result, they have to ensure their self before the program execution. All the stages of software verification, validation and testing are dependent upon the safety risk which arises from the automated system during execution [5]. In the case of large measures, the process of software validation is a process of developing the confidence level so that the software meets all the specifications and expectations. However, for an automated system, it is essential to check whether all the specifications of that system are meeting their target or not [2].

2.2. Importance of software testing. Software development undertakes the most important role of the software development process which is used to ensure the prevalence of any discrepancy in the entire software development life cycle. Software testing includes all the processes of software validation as well as verification where verification makes the product right and validation makes the right product [8]. Software testing is primarily used to ensure or verify the automated software system. In case of a software development system, testers undertake a proper software development process through which they could be able to ensure their entire development system as well as the coding system. The main purpose of the software development process can be defined as: all the developers are able to identify all the issues related to the different specification of their software system. In the case of software testing, all the developers include the process of validation through which they also could be able to provide a reliable and secure software system. Moreover, through this validation process, they could be able to build a proper structure on the upcoming issues related to their software system.

On the other hand, the software testing process also includes a verification technique through which all the developers are able to identify each issue related to their existing system. This verification system involves the process through which they could be able to verify whether the software system is meeting all the requirements [9]. Through the software testing process, most of the testers are able to ensure their self about the speed, stability, scalability of their application. The verification process is also used to check the system is following their command or not in an automated system. The process of software testing involves three different stages, including preliminary testing phase, testing phases and user acceptance testing phase [10].

All the phases mainly focus on stages of development, checking requirements, verification process, and validation process. Using this kind of testing process developers are able to establish a secure and reliable development system through which they can provide a company software for their individual system. However, software development involves three different phases in case of testing phase including verification, validation, and testing. The testing phase is specified as a separate phase which is mainly based on the perspective of the testing team.

2.3. The need for software validation and verification. Validation and verification is a whole life cycle process of the software development process through which every developer could be able to access all the issues and requirements related to their software system. The main principle of this process can be defined as, discover the defects in the

existing system, and check the ability of the system in the operational condition. All the changes in the software development system depend upon the software functions, user's expectations and market environments.

Software validation is the primary stage of the software development process by which any developer could be able to ensure their system program and prepare the system for the upcoming modifications [6]. Depending upon this process, all the organizations like IT organization can gather all the information about their customers' requirements and market environments.

In order to maintain all the functions of the software development process, it is essential to maintain a proper security management process through which every tester, as well as the developer, could be able to secure their different stages of software development including database, server, resources, and so on. In addition, through the validation process, any developer could be able to provide proper and reliable specifications for their users. In case of software development organization or team, it is essential that they have to implement Identity Documents Act 2010, through which they can verify all the document related to their users and secure the entire system [11]. It can be analyzed that, validation process in an essential part of the management of software because through the process developers are capable of making proper software to the users.

The number of failures experienced during testing can be calculated using the following equation:

$$\Delta\mu = \frac{V_0}{\lambda_0}(\lambda_p - \lambda_F)$$

where $V_0 = 500$ because one fault leads to one failure, $\lambda_0 = 2$ failures CPU hr, $\lambda_F = 5$ failures 100 CPU hr, $\lambda_p = 0.05$ failures CPU hr.

On the other hand, through a verification process, any developer could be able to improve their different issues or specification related to the existing software system. In the case of software verification, this process involves software inspections and software testing. Through this program, every tester or developer could be able to identify each issue and requirements which are the essential part of software development. It also can be analyzed that, software verification is the most important process of software development. In the following equation it is taken as the notation to denote that 0.05 failures have occurred per hour.

As a result, it can be stated that software validation is the most important process for making proper software; on the other hand, verification is the process whereby the developers of the software are able to implement the system, depending upon the requirements.

2.4. Difference between software verification and validation. In the case of a software development system, verification and validation are the most essential part of the software testing process. In order to provide a vivid distinction between validation and verification, one can entirely base on the role of specification. The main objective of validation is to ensure if the specification of the software captures the customer's needs [3]. On the other hand, the verification objectives are identified as the developed software system will meet all the requirements and also check for further updates or implementation. Verification mainly focuses on evaluating the product development process through which they could be able to verify whether the system meets all the specified requirements. On the other hand, the validation process mainly involves at the end of this program to check whether the software system meets the customer's requirements as well as expectations or not [12]. The verification process involves reviews, inspections, and meetings as objectives of this process. On the other hand, the validation process includes, white box testing, black box testing, and grey box testing as objectives of this process. However, it can be stated that the verification process involves before the execution and validation process involves after the execution of the program.

2.5. Best practices for the validation and verification process. In order to maintain all the services of the validation and verification process, every organization has to focus on their difference observation process through which every developer could be able to specify all the issues individually. Moreover, in order to maintain this process, every developer has to build a proper observation system for an automation system [9]. As a result, they could be able to observe the entire system on a regular basis. In the case of software development, verification process comes up before the execution; as a result, all the developers can start the entire system with proper planning as well as proper safety for the system. On the other hand, the validation process involves the execution of the program [13].

As a result, the entire tester could verify the entire process to ensure that whether the program meets all the requirements or not. As a result, it can be stated that the validation process is a more important process than the verification process through which all the developers are able to improve their system performance.

3. Research Methods. This research is preceded by the secondary data collection process. Positivism philosophy is undertaken so that the knowledge regarding differences in the validation and verification process of software is gathered via practical phenomena and relations. The data is interpreted through proper reason and logic by means of positivism philosophy. The deductive research approach is chosen for the research to analyze all the available evidences from online resources from a theoretical perspective. Secondary data is collected from Google Scholar, NCBI and PUBMED database. Based on the collected data, themes are set on the research topic highlighting the different testing procedures of validation and verification and analyzing the differences between them. Axial codes are formed from the themes extracting the keywords and those are further analyzed. An inclusion and exclusion diagram is provided that mentions the use of the journals and databases required for the study.

In the case of the concerned research work, the deductive research approach is used. This approach is helpful to gather authentic data based on theoretical findings. Besides, the deductive research approach is helpful to initiate research by means of pre-formulation of questions. In this respect, this approach is chosen due to evaluating and analyzing the verifications and validations of software. In order to take this approach into consideration, the theoretical analysis and evaluation would be easier.

4. Results.

4.1. Application. The difference between the validation and verification processes depends mainly on their application. While validation mainly involves the execution of the program, the verification process is involved before the execution of the program. As a result, it can be stated that through the validation process the tester could be able to identify all the issues related to the system. On the other hand, through the verification process, developers can evaluate software development in a proper way.

Moreover, the V-model also differs with each other depending upon their objectives. Validation involves three different types of box testing processes. On the other hand, verification includes three different observation processes of the software [7]. In terms of their execution, verification is a manually checking process including checking documents and files through which the developers could check the user's requirements. Conversely, validation checks the existing and the implemented program based on the specified requirements related files and documents.

In terms of the development cost, software verification is a cheaper process than the validation process which explains why most of the companies mainly focus on the software verification process maintained by the developers. However, most of the software developers are involved plans, design specifications, code, requirement specification, test cases

for the software verification process [14]. On the other hand, all the testers are involved software under testing process, through which they could be able to identify all the issues related to their system and ensure their service. On the other hand, the verification process is mainly used to explain whether the outputs are totally depending upon the input or not.

4.2. Functionality of the software system. Validation is used to provide a proper description of whether the software system is useful for the user or not. Consequently, in terms of the execution of a program, it can be stated that validation is a more important process than the verification process in the software development life cycle.

It also can be concluded that, in the context of software development process, validation and verification both are the essential part [11]. It is helpful to check the authenticity of the concerned software. As a result, all the developers, as well as the testers, are able to improve their services by adopting both processes. In the case of the IT industry, it is essential to adopt three implementations of the software system, which includes the software testing process, the software validation process, and the software verification process.

It is worth mentioning that in order to maintain a proper development process, the IT industry, in particular, has to follow a proper security testing process. Consequently, both the developers and the testers are able to stop unauthorized access to their stored data and resources.

4.3. Structure of the automated system. In order to maintain all the structures of the automated system, they have to care about the verification process. The reason behind this is that it is too complex to observe the entire program after the execution on a regular basis [7]. As a result, most of the companies are undertaking an automated software testing process, through which their developers could be able to identify all the issues in a proper way.

5. Analysis of the Secondary Research.

5.1. Difference between the validation and verification depending upon objectives. In case of verification, all the objectives are mainly used to verify the product is being developed based on the requirements or not. In order to maintain this process, verification involves three different observation processes which include meeting, reviews, and inspection. On the other hand, software validation process includes three different box testing processes, which includes white box testing process, gray box testing process and black box testing process [9].

All the box testing processes are mainly used to describe a concept on, whether the system meets the user's requirements and specifications or not. As a result, the testers and the developers will be able to build a proper concept based on their software requirements and its implementation process. It can be analyzed that, when it comes to objectives, the software validation process is more important than the software verification process [10]. The reason behind this is that through this process they could be able to identify all the issues related to their software development.

5.2. Identifying the most significant process between software validation and verification. Validation and verification have the potential to establish proper confidence through which all the software developers are able to fit that software for the purpose. Depending upon the difference between the validation and verification process, it can be stated that validation includes software testing process where verification includes software inspection process. However, in the case of software testing process, there are two types of testing process which includes validation testing and defect testing [15]. The software verification process includes the defect testing process through which they could

be able to build a proper concept of the significance of their software system. On the other hand, software validation testing process includes all the processes of their software testing after the execution of a program. As a result, it can be stated that, in the case of software development process, the verification process is more significant than the validation process [16]. The reason behind this is that the verification process includes the defect testing and debugging process through which all the developers could be able to debug their program or system after the execution of their program.

5.3. Equation for software verification. One of the most positive points on software verification process can be defined as, debugging process is used to identify the position of errors and through defect testing, they could be able to overcome those issues or errors [17]. It also can be stated that most of the companies are recently focusing on the software verification process so that they could maintain their software system in a proper way.

Unit Testing in R

Example calculates roots of quadratic equation $x^2 + px + q$

```
roots_quadratic_eqn <- function(p, q)
{
  if (!is.numeric(p) || !is.numeric(q)) {
    stop("Wrong input format: expects numeric value")
  }
  return(c(-p/2 + sqrt((p/2)^2 - q),
          -p/2 - sqrt((p/2)^2 - q)))
}
```

FIGURE 1. Quadratic equations to verify the functioning of software for unit testing

5.4. Evaluating the costing between software validation and verification process costing of V-model. In the case of an industrial software system, it is essential to focus on the costing process through which they could be able to maintain their software system in a lower budget. In software development, verification process is mainly used by the software developers to identify the requirements of their users and implement their system which is a low budget process. On the other hand, validation process is used to check whether the system meets their requirements or not which is a high budgeting process [18]. As a result, in recent years every organization mostly for IT organizations is mainly focusing on the verification process to save their cost as well as time. It also can be stated that through using the software verification process they also could save the manpower which is the most effective part for an IT organization.

5.5. Significance of validation and verification in increasing software efficiency. In the case of software efficiency, all the developers mainly focus on the debugging and defect testing process. For the development among the software system, most of the IT companies are mainly targeting their software validation and verification process through which they could be able to improve their efficiency [19]. The software verification process is mainly used to check the various performances related to their software, which are total causes for software efficiency.

On the other hand, software validation process is mainly used to focus on the various checking processes related to the software performance, which is mainly used to improve the efficiency of their software system. As a result, it can be stated that verification process is used to check the software efficiency where validation process is used to improve the software efficiency [15]. However, it can be analyzed that, the process of software

validation is more effective in case of improving the efficiency of their entire system. As a result, most of the IT firms are including a proper automated validation process through which they could improve the efficiency of their software system.

6. Conclusion. The process of validation and verification totally depends upon the source code, through which all the developers could be able to improve the performance of their software system. In addition, the software development life cycle is known as the most important structure which is used to describe software development.

There are various activities related to software development which are involved by SDLC. Those activities include architecture, software testing, specifications, debugging, maintenance and deployment. In this structure, software testing, software validation, and verification undertake the most important role for the software development process. In addition, an effective validation and verification process provides timely visibility into the readiness and quality of the system. However, depending upon the different objectives as well as the purpose of validation and verification, it can be stated that, both processes are essential and balancing each other in a proper way. Moreover, both processes are used to find an error or issue in various way, where verification is used to locate the issues in requirements and validation is used to find all the issues in software efficiency. Most of the researchers are still working on the various structures of validation and verification, through which any organization could be able to improve the efficiency of their software system. In recent years, most of the IT sectors are using a proper automated system through which they could involve both, verification as well as the validation process. As a result, those organizations are able to reduce the complexity of the software development process and improve the efficiency of their software product in a lower cost and time.

REFERENCES

- [1] D. Beyer, M. Dangl, D. Dietsch, M. Heizmann and A. Stahlbauer, Witness validation and stepwise testification across software verifiers, *Proc. of the 10th Joint Meeting on Foundations of Software Engineering*, pp.721-733, 2015.
- [2] J. Bonior, Z. Hu, T. N. Guo, R. C. Qiu, J. P. Browning and M. C. Wicks, Software-defined-radio-based wireless tomography: Experimental demonstration and verification, *IEEE Geoscience and Remote Sensing Letters*, vol.12, no.1, pp.175-179, 2015.
- [3] legislation.gov.uk, *Identity Documents Act 2010*, <http://www.legislation.gov.uk/ukpga/2010/40/contents/enacted> [Accessed 14/11/2018], 2018.
- [4] A. Arcuri and L. Briand, A Hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering, *Software Testing, Verification and Reliability*, vol.24, no.3, pp.219-250, 2014.
- [5] M. Brain, D. Kroening and R. McCleary, *Algebraic Techniques in Software Verification: Challenges and Opportunities*, pp.8-12, 2016.
- [6] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, F. J. García-Peñalvo and E. Tovar, Project managers in global software development teams: A study of the effects on productivity and performance, *Software Quality Journal*, vol.22, no.1, pp.3-19, 2014.
- [7] O. Pedreira, F. García, N. Brisaboa and M. Piattini, Gamification in software engineering – A systematic mapping, *Information and Software Technology*, vol.57, pp.157-168, 2015.
- [8] D. Beyer, Software verification and verifiable witnesses, *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp.401-416, 2015.
- [9] F. de Assis Santos and C. A. V. de Azevedo, The assistant software version 7.7 and its use in the analysis of experimental data, *African Journal of Agricultural Research*, vol.11, no.39, pp.3733-3740, 2016.
- [10] M. Dobrescu and K. Argyraki, Software data plane verification, *Communications of the ACM*, vol.58, no.11, pp.113-121, 2015.
- [11] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Peirez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J. H. Prinz and F. Noè, PyEMMA 2: A software package for estimation, validation, and analysis of Markov models, *Journal of Chemical Theory and Computation*, vol.11, no.11, pp.5525-5542, 2015.

- [12] M. D. Harrison, P. Masci, J. C. Campos and P. Curzon, Verification of user interface software: The example of use-related safety requirements and programmable medical devices, *IEEE Trans. Human-Machine Systems*, vol.47, no.6, pp.834-846, 2017.
- [13] S. Pearson, J. Campos, R. Just, G. Fraser, R. Abreu, M. D. Ernst, D. Pang and B. Keller, Evaluating and improving fault localization, *Proc. of the 39th International Conference on Software Engineering*, pp.609-620, 2017.
- [14] W. Nowatzke and E. Woolf, Best practices during bioanalytical method validation for the characterization of assay reagents and the evaluation of analyte stability in assay standards, quality controls, and study samples, *The AAPS Journal*, vol.9, no.2, pp.E117-E122, 2007.
- [15] Z. Qi, F. Long, S. Achour and M. Rinard, An analysis of patch plausibility and correctness for generate-and-validate patch generation systems, *Proc. of the 2015 International Symposium on Software Testing and Analysis*, pp.24-36, 2015.
- [16] J. Bertrane, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné and X. Rival, Static analysis and verification of aerospace software by abstract interpretation, *Foundations and Trends® in Programming Languages*, vol.2, nos.2-3, pp.71-190, 2015.
- [17] C. T. Viswanathan, S. Bansal, B. Booth, A. J. Destefano, M. J. Rose, J. Sailstad, V. P. Shah, J. P. Skelly, P. G. Swann and R. Weiner, Workshop/conference report – Quantitative bioanalytical methods validation and implementation: Best practices for chromatographic and ligand binding assays, *The AAPS Journal*, vol.9, no.1, pp.E30-E42, 2007.
- [18] Y. J. Choi, J. S. Yoo and C. L. Smith, *RELAP-7 Software Verification and Validation Plan: Requirements Traceability Matrix (RTM) Part 1 – Physics and Numerical Methods*, No. INL/EXT-15-36684, Idaho National Lab. (INL), Idaho Falls, ID (United States), 2015.
- [19] J. L. Hicks, T. K. Uchida, A. Seth, A. Rajagopal and S. L. Delp, Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement, *Journal of Biomechanical Engineer*, vol.137, no.2, 2015.