VERIFICATION AND VALIDATION OF A SOFTWARE: A REVIEW OF THE LITERATURE

Atica M. Altaie[1]
[1] *College of Computer Science and Mathematics University of Mosul, Mosul, Iraq*
atica_altaie@uomosul.edu.iq

Rasha Gh. Alsarraj[2]
[2] *College of Computer Science and Mathematics University of Mosul, Mosul, Iraq*
rasha.alsarraj@uomosul.edu.iq

Asmaa H. Al-Bayati[3]
[3] *College of Computer Science and Mathematics University of Mosul, Mosul, Iraq*
asmashade77@uomosul.edu.iq

Abstract **- With the development of the Internet, making software is often essential, also it is complicated to succeed in the project's development. There is a necessity in delivering software of top quality. It might be accomplished through using the procedures of Verification and Validation (V&V) via development processes. The main aim of the V&V has been checking if the created software is meeting the needs and specifications of clients. V&V has been considered as collections related to testing as well as analysis activities across the software's full life cycle. Quick developments in software V&V were of high importance in developing approaches and tools for identifying possible concurrent bugs and therefore verifying the correctness of software. It has been reflecting the modern software V&V concerning efficiency. The main aim of this study has been retrospective review related to various researches in software V&V and conduct a comparison between them.**

In the modern competitive world related to the software, the developers of software must be delivering on-time quality products, also the developers should be verifying that the software has been properly functioning and validating the product for each one of the client's requirements. The significance of V&V in the development of software has been maintaining the quality of software. The approaches of V&V have been utilized in all stages of the System Development Life Cycle. Furthermore, the presented study also provides objectives of V&V and describes V&V tools that can be used in the process of software development, the way of improving the software's quality.

Keywords - Software tools, Software Quality, Software Verification and Validation, Software testing, System Development Life Cycle.

I. INTRODUCTION

V&V of software is considered an important field of software engineering to develop software of high quality. Also, it is part of computer engineering departments as well as curriculum-related to universities' software [1]. Verification will be ensuring that the software has been aligned with its requirements, meeting certain needs, performance, as well as completeness based on. It is striving for fulfilling consistency, correctness related to the program translations in addition to the behavioral correctness. Validation used to determine the software's correctness at each one of the stages related to the development cycle [2].

Software's V&V standards are establishing solid frameworks to develop quality software. The main features related to V&V as an efficient standard have been the broad system's engineering method to ensure the quality has been developed in software throughout each one of the software life cycle stages [3]. Thus, the software engineering necessitates that the software must be verified throughout each one of the phases related to the development life cycle as well as validating in the case when transferred [4]. Such activities have been referred to as 'Software V&V' checking software against its specifications. Each one of the projects should be verifying and validating the software that it is producing. V&V must be establishing certainty that the software is used properly [3]. There have been a lot of conditions related to software errors which were verified for being very costly, embarrassing, and often dangerous. Optimum protection against these errors has been for validating the correctness and improving the total quality related to the software before being utilized. A significant part related to software development has been determining the software program's correctness. V&V, Software Quality Assurance (SQA), as well as Testing have been main approaches utilized for accomplishing this [4] [5].

V&V broadly test and analyze the software for determining that it is performing its required functions properly, as well as measuring its reliability and quality. V&V has been considered as a system-engineering subject for evaluating software in the system context. Compared to system engineering, it is applying a structured method for analyzing and testing software against all the functions of the system and against the user, hardware as well as software interfaces [3]. Also, they have been corresponding to each other not substituting each other. Adequate V&V might be ensuring the software's quality in addition to the quality related to processes utilized to develop and test software that is going to ultimately allow accomplishing the business objectives. Just adequate verification with no validation and conversely might not be ensuring optimum software products. Yet, it requires a combination of the two [6].

They are the two significant aspects related to software quality management. Furthermore, the verification provides answers to the questions in which software has been created in an adequate approach and the validating provide the answers if the adequate software has been created. Verification denoting precision, while the validation indicates value related to final or end product. V&V has been a significant phase utilized in a

lot of procedures in various industries. The main aim of V&V has been providing assessments related to the software ability for meeting its requirements as well as satisfying the requirements of user verification indicates to evaluating the conformance of software to its specifications, whereas the validation indicates guaranteeing that the software fulfilling the expectations of the customers [7]. The processes of V&V are providing evaluating related to the software products as well as the processes during the life cycle of software. Such evaluation showing if the system and software requirements have been precise, testable, accurate, complete, as well as consistent. The other aims of achieving V&V have been [8]:

1) Facilitating early detections and corrections of the software errors.

2) Enhancing the management insights in to process as well as product risks.

3) Supporting the processes of the software life cycle for ensuring compliances with the program's schedule, performance as well as budget requirements.

The main aims of this study are as follows:

1) Providing summarization related to modern software V&V technologies and increasing the visibility related to latest software verifiers,

2) Establishing repository related to the software V&V tasks which have been publicly provided for free use as the standard benchmark suite concerning assessing the verification software,

3) Establishing standards that facilitate comparing various V&V tools, involving property language as well as formats for results, accelerating the transfer related to novel technology of V&V to industrial practices.

Section two of this study provides the definitions related to V&V. Section three providing the role of V&V. Section four will be providing the differences between them. Section five is providing an explanation related to the tools of V&V. Section six will be providing the main conclusions.

II. COMMON DEFINITIONS FOR V&V

Verification has been related to the process's correctness. Verification is determining if the software related to certain phase system developments life cycles satisfying the requirements developed throughout former stage or not. Verification methods attempted for identifying the product errors or faults, which result in failure [9] [10]. Validation has been related to the product's correctness. Validation has been the process to evaluate the software at end related to its development for ensuring that it has been free of failures as well as complying with its requirements, also for ensuring that the product meeting the client requirements, also the specifications has been adequate. Thus, validation ensuring that the software is properly working for the correct inputs as well as giving error messages to the wrong inputs [6][11][12].

This section will be examining different definitions related to terms in associated literature and standards.

From IEEE-1012-2012 (IEEE Computer Society, 2012):[13]

Verification has been defined as the procedure to evaluate component or system for assessing if the product related to certain stage satisfying the conditions at the phase's start.

Validation has been defined as the process to evaluate component or system at the end or throughout development process for determining if it is satisfying certain requirements.

ISO/IEC 15288 (2015) providing these definitions related to verification and validation of the system:[14]

• *Validation:* it can be defined as the set of activities guaranteeing that the system has the ability of accomplishing its objectives, goals, as well as intended use.

• *Verification:* it can be defined as the set of activities which compare products related to the life cycle of the system against needed characteristics of the products. This might involve, yet isn't restricted to, certain requirements, design description in addition to system itself.

INCOSE SE Handbook, v4, build on definitions related to validation and verification utilized in ISO/IEC 1528 as well as adding certain notes as follows:[15]

• *Verification:* this is defined as set of activities comparing system element or system against the needed characteristics.

• *Validation:* this can be defined the set of activities guaranteeing that the system has the ability for accomplishing its objectives, goals, as well as intended use (meeting the requirements of stakeholders) in intended operational environments.

ISO/IEC/IEEE 29148:2011 (2011) repeating ISO/IEC 15288 definitions related to validation and verification and providing these definitions of the requirements of verification and validation:

• *Requirements validation:* defining adequate system planned via stakeholders.

• *Requirements verification:* defined as the set of requirements were indicated for ensuring characteristics related to optimum requirements have been conducted.

For meeting the requirements of stakeholders verified and validated [requirements should be complete (nothing absent which is requiring for satisfying needs or parent requirements), also correct (with no errors, such as being enough for satisfying the needs or the parent requirements) [16]. Figure1 showing such relations for requirements as follows:
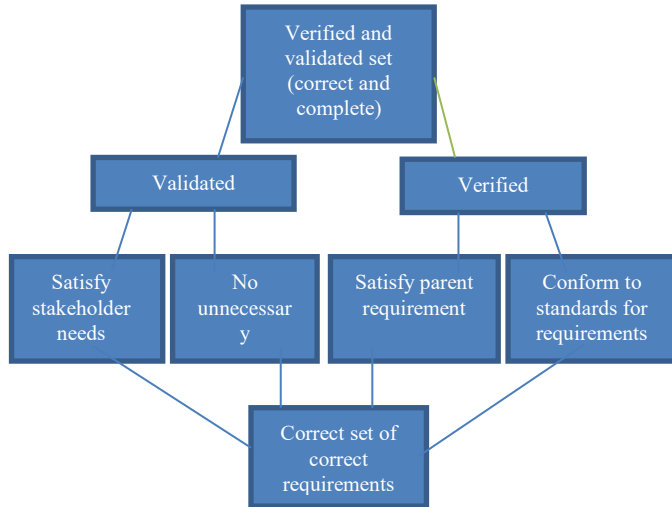
Fig.1 Hierarchical relations related to validated and verified requirements [16].

III. THE ROLE OF V&V IN SOFTWARE SYSTEM DEVELOPMENT LIFE CYCLE

The V&V's role with regard to each one of the products should be assesses through project-by-project foundation. Such assessment is going to be impacted through criticality related to products, their constraints, as well as complexity. Generally, the aim of V&V function has been ensuring that the product is satisfying the requirements of users. Therefore, everything in the specifications and requirements of products should be target regarding certain activity of V&V. For the purpose of limiting the scope of such module, yet, the approaches of V&V specified will be concentrating on functional as well as performance portions related to specifications and requirements in terms of security, maintainability, portability, safety, usability, as well as serviceability, even though being of high importance to a lot of systems, it won't be discussed here [9][17]. Verification come with the aim of "building the system right", and validation come with the aim to "building the right system." Therefore, verification examine problems like guaranteeing that the system's knowledge has been adequately specified, whereas the validation examine processes for ensuring the system is making correct decisions. V&V is of high importance in implementation and development of case-based systems. In the case when the system isn't verified, there might be errors in case representations. In the case when system hasn't been validated, then it might not be making the required quality of decisions [3] [12].

V&V in software development life cycle will be presented in this section. Particularly, what is checked with regard to all phases of life cycle as well as who is achieving such checking. Also, the approaches utilized for performing checking. Furthermore, the software V&V use reviews, testing, as well as analysis approaches for determining if the software system in addition to its intermediate products are complying with the demands. Such demands are including quality attributes as well as functional capabilities. Also, the major task of V&V has been managing the project risks through monitoring and detecting errors during the maintenance and development processes. Due to the fact that it is impractical to detect and solve all the errors early in the life cycle of a project, it has been the task of V&V contractor for identifying errors as early as possible as well as tracking progress toward resolution. Also, verification ensuring that the system (personnel, hardware, software, and documentation) are in accordance with processes and standards of organization, based on review or non-executable approaches. Furthermore, the validation physically ensuring that the system is operating on the basis of a plan through executing system functions via series of tests which might be assessed and observed. [9][17]

The life cycle of products being with the definition which specify the anticipated system's behavior. After that, the solution will be developed, built as well as tested for comparing the behavior with the initial description. In the case when it passes, then it will be utilized; however, when it does not pass, then it will get reworked till it passes. Furthermore, the lifecycle artifacts have been by-product related to such evolution[18]. Such artifacts are representing the system at specific phases of the life cycle. Requirements are representing behaviors, designs are representing solutions, while the source codes representing implementations, also tests representing qualifying argument with regard to the deployment. V&V can be considered as a process in itself, also it has its own lifecycle. The lifecycle of V&V will be run in parallel with the development. For instance, since behavior has been defined and the by-product created (requirement specification), the V&V will be performing requirements analysis. On the basis of their evaluation, the process of V&V is going to gain understanding related to the behavior of the system [3][19]. Figure2 showing continuous V&V process in the software System Development Life Cycle [9].
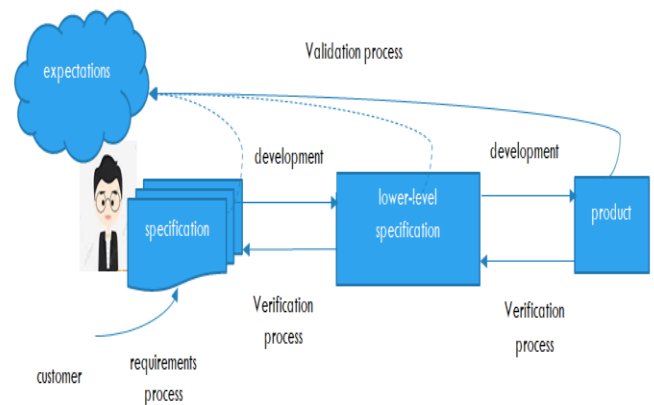

Fig 2: Continuous V&V process in software System Development Life Cycle [9][19].

IV. DIFFERENCE BETWEEN VALIDATION AND VERIFICATION

Such 2 terms have been extremely confusing for the majority of individuals, who are interchangeably using them. The table I will be highlighting differences between validation and verification.

TABLE I
DIFFERENCE BETWEEN VALIDATION AND VERIFICATION

Verification	Validation
Static process to verify program, design, document, and code[20].	Dynamic process to validate and test actual products[20] in source codes [20].
Activities such as walk-throughs, collect reviews, inspections for verifying develop[21].	Activities such as executing develop against requirements [21].
Apply approaches like meetings and review to assess specifications, codes, documents, plans, as well as requirements [22]	Applying approaches such as gray box testing and black box testing. [20]
Involve achieving right things up front in software development project – utilizing excellent practices related to deployment, design, requirements, construction, analysis, also monitor and ensure auditable workflows throughout[22].	Require involvement from all the stakeholders in specifications related to requirements as well as the throughout development [22].
Check if a software is conforming to specifications. [20].	Check if software is conforming to customer's requirements and expectations. [20].
Considered as low-level exercise, thus it might be catching errors which validation has no ability of catching. [20].	Considered as High-Level Exercise, it might be catching errors which verification has no ability of catching [20].
Considered as objective process which is not subjective decision must be required for verifying develop[14].	Considered as subjective process, also involve decisions on the efficiency of the system.[14]
Ensuring that a system meeting all functionality. [21]	Ensuring that functionalities meeting required behaviors. [21]
Code has not been executed[20].	Code has been executed[20].
It is generally come first-done prior to validation[20].	It is generally following the verification[20][22].

V. TOOLS AND APPROACHES OF V&V

- In the field of modeling:

In 2003, Engels et al presented a method to verify the characteristics of the unified modeling language UML models through the application of model checker Failures-divergence-Refinement FDR and validated the test models. So, The testing may be lifted later on the level of the model through the provision of the methods of the derivation of an implementation from UML system model as well as its testing model, taken under consideration a specific strategy of the testing. The results shown that tool suite is enabling the verification and validation of the property through the provision of the predefined partial translation types to verify and test validation strategies.[23]

In 2010, Kezadri and Pantel presented main elements related to ontology formalizing part regarding knowledge of the behavioral modeling as well as related V&V approaches. It is summarizing entities, concepts, and objects which age been indicated to be present in such area of interest as well as the relation between them. The study suggested classification related to various modeling formalisms, representations related to likely V&V approaches. System are specified utilizing many views in accordance with many modeling languages, also the properties might be evaluated with V&V [28].

In 2013, Delmas et al defined a generic process of validation and verification for the design of the model and creation of the instance which has been centered on the class diagrams of the UML and its declarative constraints. this study illustrates the way by which the formal approaches may be utilized during the variety of the phases of the typical MDE (Model-Driven Engineering) procedure, from problem formalizations to solution syntheses, and verification of the solution [30].

In 2016, In this paper primary objectives lay in making Systems engineering experts capable to create sufficiently formal Domain Specific Modeling Language DSMLs able to model a system ensuring its V&V activities. So, this approach aims to formally defining and promoting the notion of dynamic semantics for DSMLs, allowing the ability to simulate created models by using simulation and model checking techniques independently from transformation rules and mechanisms. So, the paper provided a tool-equipped method called "xviCore" allowing then system engineers to create their proper sufficiently formal DSMLs able to achieve V&V objectives and to create models that can be simulated and taken as input to a property proof process [31].

In 2017, Nastov et al provided a model to combine main four complementary strategies for verification and validation which are formal proof, guided modeling, simulation, and model appraisal. Although execution and combination of these strategies are time consuming, problematic and expensive, this study presents a tool-equipped to easy execution and combination of the V&V strategies in a tool called xviCore and introduces an acceptable level of trust in decision-making for customers. The xviCore tool is an exciting alternative to aim designers to stay independent and well-organized, to model with lower efforts, and to obtain various results of V&V strategies [32]. In the same year, Gogolla et al introduced and provided a necessary use cases to modeling software using UML and OCL in many aspects which are model verification, validation, and exploration to assist software engineers find defects and faults in descriptions of models and furthermore increase the quality of model. Experiments are applied eight use cases and shown validator with the USE model. The approaches are valuable when draw models from the initial stage of development to the verified of model properties in the testing stage [33].

In 2018, Baduel et al presented an approach which is associated with verifying and validating The Systems Modeling Language (Sys-ML) models from an industrial point of view. The approach used Object Constraint Language (OCL) for the verification of the model and utilized from Bombardier Transportation (BT) for achieving sufficient results of the V&V and accelerating the process of the development of the system with less time consumed upon system validation and testing [36].

In 2019, Schumann and Popstojanov have presented a V&V model which has been particularly fashioned for the Model-based Software Engineering (MBSwE), reflecting the inter-relationships between the variety of the tools and artifacts on one hand, and the various tasks of V&V (such as the analysis, review, and testing) on the other hand, separating all of the V&V tasks to the three levels of model-level V&V, hardware level V&V, and code-level V&V. V&V in the Model-based Software Engineering stays a valuable, but complicated task, particularly in the case where some software parts are generated in an automatic manner from the models, whereas others are written manually [38].

- In the field of coding and industrial applications

In 2006, Sherriff and Williams presented parametric model that applies persistent record related to the practices of (V&V) utilized with the program for estimating program's defect density. Furthermore, the persistent record related to practices of V&V have been recorded as certificates that have been automatically recorded as well as managed with the code. Also, the model allows developers to handle the efforts which has been put in the V&V where all the developers might be spotting the measures which were taken for ensuring the reliability related to the piece of code and treating it accordingly [25].

In 2010, Feldt et al presented two industrial case studies of the companies in the space industry of Europe which have been following the standards of the European Cooperation for Space Standardization (ECSS) in a variety of the V&V activities. Those studies which have been reported in this paper are focused on the way the standards of the ECSS are utilized by companies, the way and the degree by which it affects their procedures and, finally, they have discussed the potential ways of forwarding so as to accomplish the objective of the creation of a more cost-effective activities framework for validation and verification for space industry [5].

In 2018, Sen et al presented an overview related to Certus which is the center of the software V&V researches which is driven via the software industry. Also, Certus was established body of knowledge, tools as well as approaches for V&V related to software systems in Norwegian public and private sectors. Also, this work presenting Certus' organizational structure and describing how the health and life of Certus has been planned as well as assesses often a research-based innovation center. Certus' health has been assessing on the basis of a lot of criteria such as gender equality, standardization efforts, publications, innovation index, as well as international cooperation [34].

In 2019, Bondarev et al tackled classification related to software verification approaches (software). Article related to current static verification approaches has been implemented, the characteristics related to approaches have been examined, also a study has been achieved on detection related to dependencies in framework related to abstract interpretation approach. Also, the article involves overview related to classification as well as dynamic software verification approaches. The dynamic approaches facilitate determining just errors occurring in the case when starting the program. Also, it has been extra productive, also modern testing approach which might be detecting considerable amount of the vulnerability in the code of program, instead of utilizing static approaches related to software testing[37].

- In the field of quality of software

In 2006, Runeson et al provided analysis of current empirical works demonstrated no clear-cut answers to questions related to what defect detection approach to select. With regard to the requirements defects, the costs related to the requirements inspections have been low in comparison to conducting inadequate requirements. With regard to the design specification defects, there are certain studies indicating that inspections have been extra effective in such stage. With regard to the code, structural or functional testing has been ranked more effective in comparison to inspections in the majority of researches. The efficiency of verification has been low; the reviewers are finding just (25-50)% of the defects with the use of inspections, also the testers indicated (30-60)% with the use of testing [24].

In 2008, Drusinsky et al introduced a model to enable the software engineers in choosing the adequate approach with

regard to formal V&V tasks. The model provided 3D tradeoff space involve coverage and costs. Therefore, there is high importance in selecting FV&V approaches which are coverage-effective and cost-effective in specification/validation dimension[26].

In 2009, Monteiro et al provided the execution and analyzing the adopting of V&V activities in the CMMI (Capability Maturity Model Integration) and ML-2 (Maturity Level 2) resulted in improving or deteriorating the life-cycle of the product. Furthermore, the impact analysis conducted through the comparison of the product life-cycle state prior to and post introducing the V & V Process Areas (PA) based on the accomplished results. In the case where result has not been adequate, some of the variations in incorporating the practices of the V&V to the CMMI ML-2 is carried out to the point of achieving the optimal results [27].

In 2012, Pasareanu and Bobaru introduced learning approaches have been increasingly utilized for improving software V&V activities. For instance, the automata learning approaches were utilized to extract behavioral models related to the software systems. The study involves five presentations. The initial 4 papers addressing the automata learning as well as presenting a lot of approaches for learning a lot of types related to automata. The last study has been of distinctive concentration, since it studies the relation between automated testing and machine learning [29].

In 2018, Cheng et al provided a practical method and an effective statistical rationale in order to evaluate overall quality of the product through V&V activities. Also, the proposed method useful into derive plans that suitable for predicting reliability requirements of the product through tables and equations. The reliability must increase a threshold with predicted value of statistical confidence and give a vision to predict parameters of acceptance criteria for the product and passing it. Results and examples shown how achieving a high level of the acceptance value [35].

In 2019, selecting of V&V approaches to accommodate quality features of ISO-25010.Therefore, the authors collected a set of V&V approaches and showed to the experts to assess how each approach implemented the features of ISO-25010. The results mentioned twenty of the best practices by more than one expert can implemented automatically or semi-automatically to increase the quality of software including many types of testing: Fuzz, Penetration, Stress, and Model Checking [39]. In the same year, Mobin et al, this paper introduced a model for enhancing planning of V&V activities and product design to increase the reliability and provided a developer with critical information about product. The suggested activities of V&V planning take into account detectability, failure modes priorities, execution sequencing of V&V activities, efficiency of each one and consequences in decreasing rate of failure and increasing detectability of failure. The goal of this model is maximize reliability based on selecting an optimal activities set of V&V and sequencing based on job shop scheduling [40].

CONCLUSION

The software product's quality is considerably regulated by the quality of the utilized procedure for creating and maintaining it. The process of the software is the collection of the methods, activities, and practices guiding the people in software production. The software V&V has a significant role during the process of the development of the software. The tasks which are carried out by the V&V software are requirement analysis, structure review, requirement tracing, code inspections, design review, and validation testing. Although the V&V is costly, it is efficient and guarantees a product of high quality for the mission-critical applications such as the missile software.

Therefore, from above it can be said that V & V are an important portion of the lifecycle of the software since they provide the only manner of judging the success and the quality of software. In the V & V, it is essential to ensure that, particular rules are being followed at the software product development time and also, ensures the fact that the developed product accomplishes the needed specifications.

Which decreases the risk which is related to any software project up to a specific level through helping in detecting and correcting mistakes and errors that are unintentionally done throughout the process of the development. Via the literature survey of numerous sources of literature as approaches that have been utilized in the present paper, it has been known that it is essential for V & V to have flexibility as a result of the changes in software requirements throughout the development of the software. This is why, in the future, we will be focused on the approaches of the V & V which are going to be helpful for the enhancement of the software process in the case where there are functionality changes of the software system requirements.

The activities of the V&V are helpful in the early detection of the errors in the process of software development, thereby decreasing the time and cost of removing the errors and producing a product with higher robustness. Although the extra cost has been incurred in V&V. In addition to that, the V&V is helpful in a more efficient software development documentation.

Generally, the verification means basics (i.e. structure) of items (i.e. requirements, design, and system) which is subject to the verification, thereby ensuring that it will meet the requirements driving the item creation, whether it rules on writing the well-formed requirements, standards and best practices (internal as well as external) on design, or the requirements on system coding or manufacturing. The Validation extends to basics (i.e. structure) to the efficiency of item communication or addressing the requirements and expectations of the stakeholders at the same time as operating in the specified operational environment. We have provided an outline of a collection of the tools which may be utilized for achieving the verification and the validation of the software.

The final assumption may be made in concerns to the general verification and validation concepts. As the system engineering best practices, tools and verification and validation activities require being performed continuously during the life-cycles of the system development. Initially, the focus has been made on the needs, after that, the focus has been shifted to design, and ultimately, to the verification and validation of the system

ACKNOWLEDGMENT.

REFERENCES

[1] Mishra D., Hacaloglu T. and Mishra A., "Teaching Software Verification and Validation Course:A Case Study", International Journal of Engineering Education Vol. 30, No. 6, 2014.

[2] Gayetri Devi S.V., Nalini C. and Kumar N. , "An efficient software verification using multi-layered software verification tool" International Journal of Engineering & Technology, 2018.

[3] Kubde P., and Sable D., "The Role of Verification and Validation in System Development Life Cycle", International Journal of Research in Advent Technology, Vol.2, No.2, 2014.

[4] Sujatha P., Vijaya S. G., Sarvottama Rao A. and T. Satyanarayana, " The Role of Software Verification and Validation in Software Development Process", IETE Technical Review, Vol 18, No. I, pp. 23-26, 2001.

[5] Feldt R. , Torkar R., Ahmad E. and Raza B., "Challenges with Software Verification and Validation Activities in the Space Industry", International Conference on Software Testing, 2010.

[6] Upadhyay P., "The Role of Verification and Validation in System Development Life Cycle", IOSR Journal of Computer Engineering (IOSRJCE), Vol. 5, Issue 1, 2012.

[7] Sommerville I., "Software Engineering", Ninth edition, Addison-Wesley, 2011.

[8] IEEE Computer society, IEEE standard for software verification and validation, 1998.

[9] Ryan M. J. and Wheatcraft L. S., "On the Use of the Terms Verification and Validation", INCOSE international symposium with permission, Vol.27, Issue.1, 2017.

[10] Beyer D. and Lemberger T., "Software Verification: Testing vs. Model Checking A Comparative Evaluation of the State of the Art", Hardware and Software: Verification and Testing. Springer, 2017.

[11] Kamalrudin M. and Sidek S. , "A Review on Software Requirements Validation and Consistency Management", International Journal of Software Engineering and its Applications, 2015.

[12] Schumann J. and Goseva-Popstojanova K., "Verification and Validation Approaches for Model-based Software Engineering", ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2019.

[13] Computer Society, IEEE-STD-1012-2012 - IEEE Standard for System and Software Verification and Validation, New York: IEEE Computer Society, 2012.

[14] ISO/IEC 15288, ISO/IEC 15288:2015, 2015, Systems Engineering—System Life Cycle Processes.

[15] INCOSE-TP-2003-002-04, 2015, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Version 4, Edited by Walden, David D., et ISO/IEC/IEEE 29148:2011, ISO/IEC/IEEE 29148 Systems and Software Engineering—Life Cycle Processes—Requirements Engineering.

[16] Carson R. S. and Noel R. A. , "Formalizing Requirements Verification and Validation", Symposium of International Council on Systems Engineering, Vol.28, Issue1, 2018.

[17] Abdulwahid M. N., "The Development of Life Cycle Technique for Software Verification and Validation", Sumerianz Journal of Scientific Research, Vol. 1, No. 1,2018.

[18] Alawneh L., Debbabi M., Hassaine F., Jarraya Y., and Soeanu A., "A Unified Approach for Verification and Validation of Systems and Software Engineering Models", 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06), 2006.

[19] Michael J. B., Drusinsky D., Otani T. W., and Shing M. T., "Verification and validation for trustworthy software systems". IEEE Software, 2011.

[20] Kumar R. and Kumar N., "The Role of Verification and Validation in System Development Life Cycle", International Journal of Engineering Technology Science and Research (IJETSR), 2015.

[21] Cardenas J., Loken R., Rodriguez J. , Arribas D., Ayala D. R., Martin J. and Mendez J. A.," IEC 61850: Believe Or Don't Believe In Testing That Is The Question!", 25th International Conference on Electricity Distribution, CIRED ,2019.

[22] Rao K. N. and Sastri A. P., "Overcoming Testing Challenges in Project Life Cycle using Risk Based Validation Approach", International Journal on Computer Science and Engineering (IJCSE),2011.

[23] Engels G., Kuster J. M., Heckel R., and Lohmann M., "Model-Based Verification and Validation of Properties" Electronic Notes in Theoretical Computer Science, Vol.82, 2003.

[24] Runeson P., Andersson C., Thelin T., Andrews A. and Berling T., "What do we know about defect detection methods?", IEEE Software, 2006.

[25] Sherriff M. and Williams L., "Defect Density Estimation through Verification and Validation", The 6th Annual High Confidence Software and Systems Conference, 2006.

[26] Drusinsky D., Michael J. B. and Shing M., "A Visual Tradeoff Space for Formal Verification and Validation Techniques", IEEE Systems Journal,2008.

[27] Monteiro P., Machado R. J., and Kazman R., "Inception of software validation and verification practices within cmmi level 2", In Fourth International Conference on Software Engineering Advances, pages 536–541, Sept 2009.

[28] Kezadri M. and Pantel M. , "First Steps Toward A Verification and Validation Ontology", In International Conference on Knowledge Engineering and Ontology Development (KEOD'10), 2010.

[29] Pasareanu C. S. and Bobaru M. G., "Learning techniques for software verification and validation", International Symposium on Leaveraging Applications of Formal Methods, Verification and Validation (ISoLA), 2012.

[30] Delmas, R., Pires, A.F., and Polacsek, T., "A verification and validation process for model-driven engineering", EDP Sciences, pp. 455-468, 2013.

[31] Nastov, B., Chapurlat, V. and Dony, C, "Towards V&V suitable Domain Specific Modeling Languages for MBSE: A tooled approach", 26th Annual INCOSE International Symposium, Edinburgh, 2016.

[32] Nastov B., Chapurlat V., Pfister F., and Dony C. " MBSE and V&V: a tool-equipped method for combining various V&V strategies", International Federation of Automatic Control World Congress (IFAC WC 2017), Toulouse, France, July 2017.

[33] Gogolla M., Hilken F., and Doan K., "Achieving Model Quality through Model Validation, Verification and Exploration" Journal on Computer Languages, Systems and Structures, Elsevier, 2017.

[34] Sen, S., Marijan, D. and Gotlieb, A., "Certus-An Organizational Effort Towards Research-based Innovation in Software Verification & Validation", International Journal of System Assurance Engineering and Management, 2018.

[35] Cheng S., Kupfer K. and Shammas S., "Optimized sampling design and rationale for verification and validation", Qual Reliab Engng Int, Willey, 2018.

[36] Baduel R., Chami M., Bruel J.-M, and Ober I., "SysML Models Veri_cation and Validation in an Industrial Context: Challenges and Experimentation", European Conference on Modelling Foundations and Applications, Springer, Cham, pp 132-146, May 2018.

[37] Bondarev S. E., Chudinov M. A., Prokhorov A. S., "The Analysis of Existing Methods of Software Verification", IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2019.

[38] Schumann J. and Popstojanov K. G., "Verification and Validation Approaches for Model-based Software Engineering", ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2019.

[39] Mendoza I., Kalinowski M., Souza U. and Felderer M.: Relating verification and validation methods to software product quality characteristics: results of an expert survey. In: 11th Software Quality Days (SWQD). Lecture Notes on Business Information Processing, Vienna, Austria. Springer, 2019.

[40] Mobin, M., Li Z., Hossein S. H. and Wub G., "An approach for design Verification and Validation planning and optimization for new product reliability improvement", Reliability Engineering & System Safety, Elsevier, 2019.