Identifying Impact Criteria for Software Requirement Verification and Validation Tools Selection

Khalid Aldousari
Dept.of Software Engineering
King Saud University
Riyadh, Saudi Arabia
442106411@student.ksu.edu.sa

M. Abdullah-Al-Wadud
Dept.of Software Engineering
King Saud University
Riyadh, Saudi Arabia
mwadud@ksu.edu.sa

***Abstract*— In the past few decades, the software requirement gathering phase as part of the software development life cycle has become of great interest to software project managers, developers, and clients. It has become a critical factor in project success and defining its quality. The requirement gathering phase is how developers gather the software projects' functional and non-functional requirements. Therefore, developers need to go through intensive testing activities to ensure that these requirements are correct and sufficient. In addition, project managers employ software tools for requirements verification and validation. However, there is a lack of research studies for identifying criteria affecting the selection of software tools used for requirements verification and validation. In this work, we focus on identifying some criteria to choose software tools for requirement verification and validation to serve the requirements gathering phases. The criteria can be utilized as a selection guide for IT employees and managers to select appropriate verification and validation tools.**

Keywords — Software V&V tools, selection criteria, styling, software development life cycle

I. Introduction

Technology has rapidly developed, resulting in increased competition; this phenomenon motivates researchers to seek new avenues of improvement in software development and its process [1]. The technique of splitting development work into discrete phases is referred to as the software development procedure [12]. This facilitates software project management and enhances the efficiency of the work. The software development life cycle (SDLC) refers to the process of performing the software development processes in the creation of software applications [2]. The requirement collection phase is the process in which developers collect the business requirements (functional) and technical requirements (non-functional) of the software project. Then in the design phase, the software is designed and built (by coding) based on the collected requirements [3]. Next, intensive testing activities are accomplished in the testing phase in order to guarantee that the developed software meets the expectations. Afterwards, the software is delivered to the end-user in the deployment phase. At last, errors, problems and improvement issues raised by different stakeholders are addressed in the maintenance phase. The first phase in SDLC is the requirements gathering process. The requirements define the stakeholder needs and expectations behind the development of a new software project, as well as how the software will act in order to meet those needs [4]. In general, requirements engineering consists of several activities, including rrequirement elicitation, rrequirement specification, rrequirement verification and validation, and rrequirement management. This area remains understudied, underlining the need for more empirical studies to illustrate the identifying criteria affecting the selection of software tools used for requirements verification and validation

Requirement extraction is the activity of obtaining knowledge associated with customer needs. Therefore, in order to collect the system requirements based on the need of the users, a number of stakeholders are selected to be contacted in elicitation activity [5]. The primary goal of requirements validation is to develop full, accurate, and significant system requirements and represent what the user requires [6]. There is an urgent need to provide more care at the phase of collecting requirements in order to ensure the correctness of their collection and to come up with software that satisfies stakeholders and their expectations, where these arrangements and care reduce the chances of re-working on collecting requirements again [7]. The process of defining if the proposed software fits the gathered requirements using techniques such as inspection, walkthroughs, and reviews is known as the requirements verification process [8]. On the other hand, the activity of assessing, recording, tracking, and prioritizing requirements, as well as handling interactions with necessary parties, is known as requirements management.

Requirements verification and validation (V&V) tools are necessary to make sure that we produce the right software. The verification tool is to ensure that software meets specifications. Validation tool focuses on whether the software meets the expectations and requirements of the end user [25]. Hence, selection of the perfect tools for requirement V&V is very important. The industry, however, mainly focuses on selection of requirement engineering tool in general based on some common criteria including whether the tool provides elicitation, analysis, specification, verification and validation, and requirements management. There are also some criteria available for selection of requirement V&V tools. For example, whether the tool provides with storing and managing requirements elicitation templates, prioritization forms, generating exception reports on verification plan cases with no verification, etc. However, the criteria are selected on brute force manner at the time of selecting V&V tools, and do not always fit the software at hand in performing requirements verification and validation [9]. In this work, we perform an investigation

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00365

about significant criteria that impact the selection of requirements verification and validation tools. We first study different criteria currently being used by organizations along with the efficacies. Then we propose a list of criteria that can be used in the process of choosing appropriate requirement V&V tools.

II. Related Works

In this section we will go through some currently available criteria selection models for V&V techniques and tools, and discuss the shortcomings to be fulfilled.

A. *Wang selection model*

Wang [30] presented a model constituting of two key aspects: a characterization approach to developing a V&V techniques catalogue that packages the available techniques together with the information about their application conditions, and a planning and tailoring strategy for project-specific selection of the appropriate V&V techniques from the established catalogue according to the goals and characteristics of a simulation study[30]. Simulation studies help make improved decisions when choosing statistical models. The reason for using the simulation study was to model real-life or hypothetical situation on a computer to solve it. Wang developed a modelling and simulation (M&S)-specific characterization model that introduced a clear step for techniques selection[30].

The characterization techniques include analyzing the available V&V method applicable to M&S applications, outlining an M&S-specific characterization model specifying the relevant properties of V&V techniques, employing the defined model to characterize the existing V&V methods and developing a catalog of the characterized V&V techniques, and selecting the appropriate V&V techniques from the V&V techniques catalog according to the goals and characteristics of an M&S project.

Different researchers have proposed multiple models that have shown some gaps. For example, the characterization approach proposed by Wang [30] cannot be utilized for selecting V & V techniques in the M&S context due to several factors:

- It is only applicable in the testing phase; however, V&V model should be performed as a continuous activity throughout the M&S life cycle.
- This characterization scheme was designed specifically for characterizing techniques for testing only. Other quality assurance techniques, such as inspections, reviews, and walkthroughs are difficult to characterize.
- The model's attributes are defined for software projects and are not always applicable to simulation studies
- The selection of more than one testing method is not possible for a specific software or project
- No process is linked with the model to support the technique selection.

These issues underline the need for an improved model to be developed for characterizing V&V techniques.

B. *Altinok et al. selection model*

Altinok *et al.* introduced a Goal-Oriented Categorization (GOC) approach as a dynamic V&V method selection strategy[32]. This model outlines two main methods for constructing the GOC model: using a default criterion, categorizing V&V methods based on their identified features; and setting a specific goal to categorize V&V methods when project management information and relevant software metrics are obtainable [32]. They performed with selection process of the simulation V&V techniques selected in accordance with the relevant project or specific criteria facilitated by the Goal-Oriented Categorization, a proposed , GOC as a V&V method selection process model that is specific to and adaptable to any simulation work.

The development of the model has three main stages:

- Categorizing V&V methods considering the default criteria.
- Retrieving simulation information based on simulation project management and software metrics.
- Generating the GOC chart and updating this structure in case of changes in the process.

Altinok et al. (2021) underlined a gap in the current models. Accordingly, it is vital to offer the accuracy and level of credibility of any simulation software by utilizing the proper techniques [32]. Most of V&V techniques have been created to be applied in the V&V process for simulation models. Thus, the technique selection process needed to perform V&V activities tends to be complex, especially when planning feasibility and cost issues for different simulation projects.

C. *Strooper & Wojcicki selection model*

Strooper and Wojcicki [31] explored several issues related to the selection and evaluation of combined V&V technologies based on personal experiences with the V&V of concurrent Java components [31]. They designed a V&V method that was systematically derived through an analysis of the possible failures that can occur in concurrent or simultaneous Java components [31]. The model combined inspection, static analysis, and dynamic testing coupled with empirical methods that utilize analysis of fault data and experiments to evaluate V&V combinations. Strooper & Wojcicki presented their ideas for an alternative technique that can be utilized to help with both the selection and evaluation of cost-effective V&V combination in a specific context [31]. Accordingly, combinations of V&V techniques are appropriate explicitly depends on the context, such as application domain and criticality of software. Strooper & Wojcicki considered different V&V technologies for concurrent (Java) components [31]. They also considered the V&V of components instead of complete applications or systems.

Strooper & Wojcicki (2007) associated a significant part of the cost and effort for development and maintenance with V&V activities involving selecting and applying a mix of V&V technologies. Regrettably, only limited is known regarding the cost-effectiveness of individual technologies or how to derive the most cost-effective combination.

D. The Missing/Gap/Extra of existing selection models

V&V of advanced control systems are necessary for their utilisation in fielded systems. Multiple studies have explored comprehensive V&V processes encompassing analysis, simulation, and experimental testing and they use in assessing closed-loop system performance and identify system limitations. Presently, there exist multiple studies describing different V&V techniques and their application. However, potential accessible findings and experiences on the effective selection of suitable V&V methods for a specific modeling and simulation (M&S) context is still missing or understudied. All the researches in the market only setting criteria for V&V techniques there is no research that tackle the criteria selection for V&V tools. In this research we will develop criteria and model for V&V tools selection.

III. Proposed Model

Based on the present research gaps, the proposed V&V selection model is a continuous activity throughout the entire M&S life cycle; thus, the V&V tool evaluation approach should facilitate systematic technique selection. The main selection criteria should include different aspects such as design documentation, operation costs, Computational, Development vs. Maintenance Activities and User Constituencies in order to cover selection of V&V tools for different phases of SDLC.

Figure 1 illustrates the proposed model for the V&V process. It aggregates the activities for performing verification and validation and leveraging the verification and validation productions. The key attributes in the proposed model include applicability and functionality.

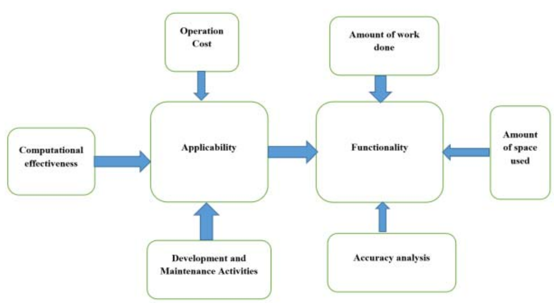

Figure 1. Proposed model for the V&V process.

Hence the objectives of the proposed evaluation model scheme can be outlined as follows:

A. Design Documentation

This involves:

- verifying that the design is consistent with, and has satisfied, the requirements
- performing consistency and completeness checks within the design itself,
- verifying the consistency of the code with the design
- The tools proposed by the evaluation model should provide:
- A thorough and complete test of the code based on the design.
- support to select appropriate techniques for each V&V activity throughout the Method & Software life cycle;
- information aiding the project management with planning the V&V
- Applicability to any simulation study with well-defined and structured model development and V&V process.

On this topic, the proposed model chose:

1) Applicability.

An analysis of an algorithm can be limited by the current state of the profession of the software developer. The Software V&V tools need to be generally applicable to the developers. The tool should be applicable for a V&V technique within the Method & Software life cycle, the information about in which V&V activities, to which artifacts of a model a technique can be applied, and which type of model errors it helps to detect. Moreover, the proposed model considers the operational conditions, whether the application of a V&V technique is coupled with a particular development paradigm of a simulation study, modeling formalism, simulation type, and simulation language; or whether an observable system (in which it is possible to collect data on the operational behavior) or execution of a model is required.

B. Operation costs

Selection of V&V techniques is a decision process to determine whether or not the application of the technique and the costs of its application match a given Method & Software context. Therefore, when constructing a Method & Software-specific characterization, the applicability of techniques and the costs of their usage are crucial and have to be included in form of several suitable attributes.

The cost to analyze an algorithm is dependent on the complexity of the software and the amount of understanding of algorithms of the same class. Usually, software development is generally a manual technique. Effective use of V V&T requires thoughtful problem and solution consideration. Tools do exist that use symbolic execution to automatically generate loop invariant assertions. The cost then becomes that of symbolic execution. On the other hand, some software V & V tools are more expensive than others.

A cost-effective procedure, therefore, is to develop intermediate assertions only for particularly important parts of the computation. Input assertions should always be employed, and output assertions whenever possible.

Proposed model considered:

Low-cost categorization: To be used in cases where there are budget constraints in human resources and financial matters.

C. Computational

Learning these tools requires significant training in mathematics and computer science. A functional understanding of tools is necessary to use them. It should not take long to sufficiently learn the specification syntax for assertions acceptable to that tool. The generation of useful assertions is necessary for this technique to be truly valuable.

Running this software should be done by a specialist. Training and experience in writing assertions are the keys to their effective use. Thoughtful consideration of the requirements contained should enable a programmer to begin with useful assertions. Experience will sharpen the ability, especially if a dynamic assertion processor or other allied technique is also used.

Here we will consider computational effectiveness in two subcategories:

- Experience-based: It is used when there is enough competence in the relevant process.
- Coverage: It is used when an extensive V&V process is required.

D. Development vs. Maintenance Activities

This phase includes definition, design, and implementation and other phases like requirements definition, analysis and design, development, validation, and implementation.

This is valid when accessing the code of the simulation software and can provide important information for the V&V method selection strategy.

These considerations can be discussed as follows:

a) Functionality.

Functionality may be discussed in these terms:

- Amount of work done - Understand that an algorithm is correct if, when given a valid input, it fulfills the requirements using the right time.
- Amount of space used - The number of memory used by a software, and the number of seconds required to execute a program, depends on the particular implementation.
- Accuracy analysis - The computational stability of the tool is verified by determining that the integrity of round-off accuracy is maintained. It is done manually at the requirements or specification level.

E. User Constituencies

User involvement and sophistication were found at each of the sites interviewed. One site had a formal group that interfaced between the user and the developer to help prepare validation plans. On two sites, users were involved to some informal degree, in both requirements specification and acceptance testing.

Hence, proposed model considered::

a) Scope.

Simplicity - Simplicity is the case and most straightforward way of solving a problem although not the most efficient. Yet simplicity in a tool is a desirable feature. It may make verifying the correctness of the software easier, and it makes writing, debugging, and modifying a program for the software easier. The time needed to produce a debugged program should be considered when choosing a tool, but, if the tool is to be used very often, its efficiency is probably the determining factor in the choice.

IV. Result

The proposed model was compared with several other V & V selection models, including the ones developed by Wang (2013), Strooper and Wojcicki (2007), and Altinok et al. (2021) among others. The proposed model's characterization technique follows a simple technique [30] [31] [32].

As illustrated in the proposed model, the main constituents include testing applicability and functionality. When testing applicability, the main focus will be operations costs, computational effectiveness, and development maintenance activities. On the other hand, functionality will focuses on amount of space used, amount of work done, and accuracy analysis. The model will focuses mostly on verifying and validating requirements. In this step, each step, each requirement is verified and validated to ensure that they are the correct requirement in terms of applicability and functionality. Consequently, it ensures that the requirements meet the overall objective of the system and all stakeholder needs. Verification and validation is done continuously throughout the development of requirements at every level and as part of baseline activities and reviewed during the system requirement reviews.

From the proposed selection model, we will have these result in table below:

TABLE I. SELECTION CRITERIA CATLOUGE FOR EACH TOOL

Category	Attribute	SDLC Phases		
		P1	*P2*	*P3*
C1	Applicability	80%	95%	70%
C2	Functionality	90%	80%	85%

V. Conclusion

There is a significant limitation today regarding the criteria selection model for V&V tool. Our proposed model will tackle such gaps through a systematic process to choose the right V&V tools for a project. We have developed criteria that can be used as a selection guide for IT employees and managers as well as clients to select the appropriate verification and validation tool. The main component of the proposed criteria include applicability and functionality of the model to select the appropriate verification and validation tool. In a broader perspective, the criteria also include other factors such as the application area, the type of processing performed and whether or not the systems were in existence or being developed, the size of projects, and the constraints affecting the projects.

REFERENCES

[1] J. Herbsleb and D. Moitra, "Global software development", IEEE Software, vol. 18, no. 2, pp. 16-20, 2001. Available: 10.1109/52.914732.

[2] Y. Tiky, "Software development life cycle.", Hongkong: THe Hongkong University of Science and Technology, 2016. [Accessed 18 July 2022].

[3] S. Yau and J. Tsai, "A survey of software design techniques", IEEE Transactions on Software Engineering, vol. -12, no. 6, pp. 713-721, 1986. Available: 10.1109/tse.1986.6312969.

[4] J. Dick, E. Hull, and K. Jackson, "System Modelling for Requirements Engineering", Requirements Engineering, pp. 57-92, 2017. Available: https://doi.org/10.1007/978-3-319-61073-3_3 [Accessed 18 July 2022].

[5] M. Yousuf and M. M.Asger, "Comparison of Various Requirements Elicitation Techniques", International Journal of Computer Applications, vol. 116, no. 4, pp. 8-15, 2015. Available: 10.5120/20322-2408.

[6] I. Johnston, "Review: Requirements Engineering: A Good Practice Guide", The Computer Bulletin, vol. 40, no. 2, pp. 31-31, 1998. Available: 10.1093/combul/40.2.31.

[7] Q. Manan, M. Asghar and S. Ghayyur, "Scaling of critical success factors for requirements engineering in the development of large scale systems", vol. 1, 2009. [Accessed 18 July 2022].

[8] W. StClair and S. StClair, "Automated management of software requirements verification", U.S. Patent No. 8,949,770, 2015. [Accessed 18 July 2022].

[9] "Requirements Verification Tools | NIST", NIST, 2021. [Online]. Available: https://www.nist.gov/itl/ssd/software-quality-group/requirements-verification-tools. [Accessed: 18- Jul- 2022].

[10] V. Axel, "Goal-oriented requirements engineering: A guided tour." In Proceedings fifth IEEE international symposium on requirements engineering, pp. 249-262, 2001. [Accessed 18 July 2022].

[11]]B. Cheng and J. Atlee, "Research directions in requirements engineering." *Future of Software Engineering (FOSE'07)*, pp. 285-303, 2007. [Accessed 18 July 2022].

[12] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap", *Proceedings of the Conference on the Future of Software Engineering*, pp. 35-46, 2000. [Accessed 18 July 2022].

[13] B. Maalem and N. Zarour, "Challenge of validation in requirements engineering", *Journal of Innovation in Digital Ecosystems*, vol. 3, no. 1, pp. 15-21, 2016. Available: 10.1016/j.jides.2016.05.001

[14] A. Terry Bahill and S. Henderson, "Requirements development, verification, and validation exhibited in famous failures", *Systems Engineering*, vol. 8, no. 1, pp. 1-14, 2005. Available: 10.1002/sys.20017.

[15] R. Mahalakshmi and R. Saranya, "Comparison of Software Requirements Tools", *Research Journal of Science and Technology*, vol. 9, no. 2, p. 272, 2017. Available: 10.5958/2349-2988.2017.00049.3.

[16] "Atlassian Marketplace", *Marketplace.atlassian.com*, 2021. [Online]. Available: https://marketplace.atlassian.com/apps/1213064/r4j-requirements-management-for-jira?hosting=server&tab=overview. [Accessed: 18- Jul- 2022].

[17] D. Iqbal, A. Abbas, M. Ali, M. Khan and R. Nawaz, "Requirement Validation for Embedded Systems in Automotive Industry through Modeling", *Ieeexplore.ieee.org*, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8949471. [Accessed: 18- Jul- 2022].

[18] A. Shah, M. Alasow, F. Sajjad and J. Baig, "An evaluation of software requirements tools.", *In 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 278-283, 2018. Available: https://doi.org/10.1109/INTELCIS.2017.8260075 [Accessed 18 July 2022].

[19] L. Filion, N. Daviot, J. Le Bel and M. Gagnon, "Using Atlassian tools for efficient requirements management: An industrial case study", *In 2017 Annual IEEE International Systems Conference (SysCon)*, pp. 1-6, 2017. Available: https://doi.org/10.1109/SYSCON.2017.7934769 [Accessed 18 July 2022].

[20] I. Zafar, A. Nazir, A. Shaheen, B. Maqbool and W. Butt, "Why Pakistani Software Companies don't use Best Practices for Requirement Engineering Processes.", *In 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 996-999, 2018. [Accessed 18 July 2022].

[21] R. Hashim, M. Abbas and M. Hashim, "Critical success criteria assessment in software projects." *In Proceedings of the 2013 Science and Information Conference*, pp. 282–287, 2013. [Accessed 18 July 2022].

[22] M. Nor Aiza, M. Kamalrudin, S. Sidek, M. Robinson and J. Grundy, "An automated collaborative requirements engineering tool for better validation of requirements.", *In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pp. 864-869, 2016. Available: https://doi.org/10.1145/2970276.2970295 [Accessed 18 July 2022].

[23] Y. Lee, H. In and R. Kazman, "Customer requirements validation method based on mental models.", *In Proceedings of the 21st Asia-Pacific Software Engineering Conference*, vol. 1, pp. 199-206, 2014. Available: https://doi.org/10.1109/APSEC.2014.39 [Accessed 18 July 2022].

[24] P. Fenkam, H. Gall and M. Jazayeri, "Visual requirements validation: Case study in a CORBA-supported environment.", *In Proceedings of IEEE Joint International Conference on Requirements Engineering*, pp. 81–88, 2002. Available: https://doi.org/10.1109/ICRE.2002.1048508 [Accessed 18 July 2022].

[25] F. Yousuf, Z. Zaman and N. Ikram, "Requirements validation techniques in gsd: A survey", *In 2008 IEEE International Multitopic Conference*, pp. 553-557, 2008. Available: https://doi.org/10.1109/INMIC.2008.4777800 [Accessed 18 July 2022].

[26] S. Conger, "Software Development Life Cycles and Methodologies", *International Journal of Information Technologies and Systems Approach*, vol. 4, no. 1, pp. 1-22, 2011. Available: 10.4018/jitsa.2011010101 [Accessed 2 August 2022].

[27] A. M. Altaie, R. Gh. Alsarraj and A. H. Al-Bayati, "Verification and Validation of A Software: A Review of the Literature", *Iraqi Journal for Computers and Informatics*, vol. 46, no. 1, pp. 40-47, 2020. Available: 10.25195/ijci.v46i1.249.

[28] A. Guerrouat and H. Richter, "A Formal Approach for Analysis and Testing of Reliable Embedded Systems", *Electronic Notes in Theoretical Computer Science*, vol. 141, no. 3, pp. 91-106, 2005. Available: 10.1016/j.entcs.2005.02.050.

[29] A. Mohpal, "Assessment and Recommendation of Requirement Management Tool Based on User Needs.", *Library.ndsu.edu*, 2012. [Online]. Available: https://library.ndsu.edu/ir/bitstream/handle/10365/22275/Aditi%20Mohpal.pdf?sequence=1. [Accessed: 02- Aug- 2022].

[30] Wang, Z. (2013, December). Selecting verification and validation techniques for simulation projects: A planning and tailoring strategy. In 2013 Winter Simulations Conference (WSC) (pp. 1233-1244). https://doi.org/10.1109/WSC.2013.6721511 .

[31] Strooper, P., & Wojcicki, M. A. (2007, July). Selecting V&V technology combinations: how to pick a winner?. In 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007) (pp. 87-96). https://doi.org/10.1109/ICECCS.2007.40 .

[32] Altinok, Y. K., Artuner, H., & İpek, A. H. (2021, December). Verification and Validation Methods Selection Based on Goal-Oriented Categorization for Simulations. In 2021 2nd International Informatics and Software Engineering Conference (IISEC) (pp. 1-5). IEEE. https://doi.org/10.1109/IISEC54230.2021.9672448 .