

Importance of Software Verification and Validation

Shivanand S. Rumma

H.O.D, Dept. of P.G. Studies in Computer Science,
Govt .First Grade College, Bidar, India.
shivanand_sr@yahoo.co.in

Sachhidanand

Asst. Professor, Dept. of Computer Science,
Govt. First Grade College, Aurad(B), India.
sachhi.r@gmail.com

Abstract – During the course of software development, project developers will employ several different verification and validation (V&V) practices with their software, but these efforts might not be recorded or maintained in an effective manner. Verification is the assurance that the products of a particular development phase are consistent with the requirements of that phase and preceding phases, while validation is the assurance that the final product meets system requirements. This paper provides the importance, concepts and understanding the verification and validation for the software development and gave the overview on V&V of Multi-Agent Systems (MAS) . The verification and validation techniques will help provide higher quality software and that is deliverable in the required time period to the customer or end user.

Keywords: Verification, Validation, Testing, Defects, Requirement Document, MAS

I. INTRODUCTION

Verification is the process of evaluating work-products of a development phase to determine whether they meet the specified requirements. Verification is a process of evaluating the intermediary work products of a software development lifecycle to check if we are in the right track of creating the final product.[1] Verification ensures that the product is built according to the requirements and design specifications and it also answers to the question of , ‘Are we building the product right?;. The documents which are produced during the development phases like requirements specification, design documents, data base design, ER diagrams, test cases, traceability matrix etc.

The development of software starts with a requirements document that is SRS, which is also used to determine eventually whether or not the delivered software system is acceptable. It is therefore important that the requirements specification contains no errors and specifies the client’s requirements correctly.[3] Validation is the process of evaluating the final product to check whether the software meets the business needs. Test execution which we do in our day to day life are actually the validation activity which includes smoke , functional , regression and systems testing etc.

Verification refers to the set of tasks that ensure that software correctly implements a specific function. But, the Validation refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. Boehm states another way:

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”[2]

II. V&V OBJECTIVES

The following objectives provide a framework within which it is possible to determine the applicability of various V&V approaches and techniques provided.

- 1) Performance: Product satisfies its per formance requirements.
- 2) Correctness: the product is fault free.
- 3) Necessity: everything in the product is necessary.
- 4) Sufficiency: Extent to which the product is complete
- 5) Consistency: the product is consistent within itself and with other products.

III. V PROCESS MODEL

V&V- model means Verification and Validation model. It’s just like the waterfall model and the V-Shaped life cycle is a sequential path of execution of processes. Every phase must be completed before the next phase begins. The Testing of the product is planned in parallel with a corresponding phase of development.

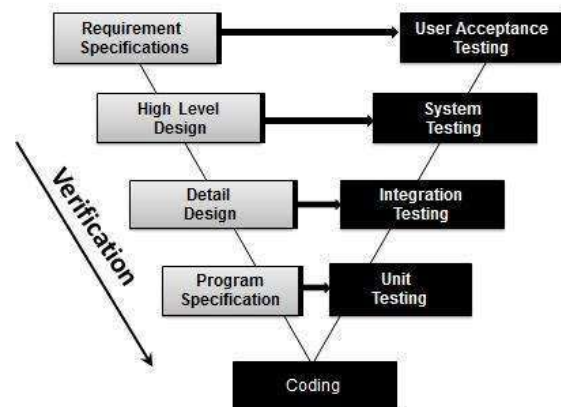


FIG 1: V PROCESS MODEL

V-Shaped Model application is almost similar with waterfall model as both the models are of linear sequential type. Here, the requirements have to be very clear and prepare complete documentation of the system before the project starts because it is usually expensive to go back and make changes.

IV. LET'S TRY TO UNDERSTAND THE DIFFERENCES BETWEEN VERIFICATION AND VALIDATION

Verification	Validation
Verification uses methods like reviews, inspections, walkthroughs and desk-checking etc.	Validation uses the methods like black box, white box testing and non-functional testing.
Verification does not involve executing the code	Validation always involves executing the code
Looks "Are we building the product right?"	Looks "Are we building the right product?"
This process includes checking documents, design, code and program	Validation is a dynamic mechanism of testing and validating the actual product
Next is, whether the software conforms to specification is checked	It checks whether software meets the requirements and expectations of customer
Verification involves all the static testing techniques	It includes all the dynamic testing techniques.
This is done without executing the software	Is done with executing the software
It finds bugs early in the development life cycle	It can find bugs that the verification process can not catch
Target is application and software architecture, complete design, specification, high level and data base design etc.	In Validation, the target is actual product.
QA team does verification and make sure that the software is as per the requirement in the SRS document.	With the involvement of testing team validation is executed on software code.
Cost of errors caught in Verification is less than errors found in Validation.	Cost of errors caught in Validation is more than errors found in Verification.
Verification comes before validation	Validation comes after verification

Table 1: Comparison of V & V

V. METHODS OF V AND V

- a) Methods of Verification
 - 1. Walkthrough
 - 2. Inspection
 - 3. Review
- b) Methods of Validation
 - 1. Testing
 - 2. End Users

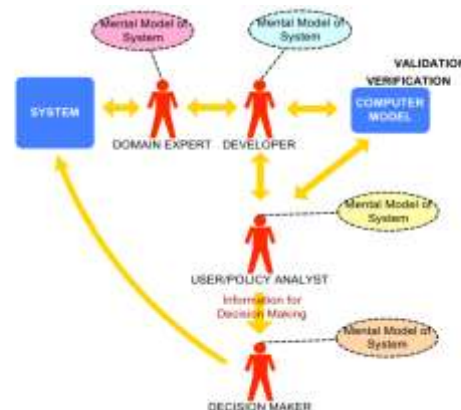


Fig. 2: An Information Focused View of Model V&V

VI. V & V OF MULTI AGENT SYSTEMS

MAS (Multi-Agent Systems) are used as a paradigm for conceptualizing, designing, and implementing software engineering systems that are beyond the individual knowledge of the traditional centralized approaches. During the V&V process, MAS is subjected to a several testing forms to perform either verification or validation or V&V.[4]

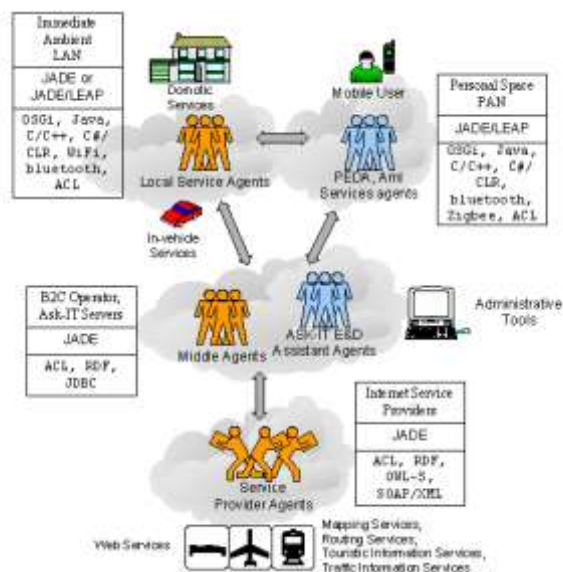


Fig.3 : Multi Agent System Example [4]

VII. LIMITATIONS OF V AND V

The following are the several theoretical and practical limitations that make this objective impossible to obtain for many products.

- 1) Firstly, theoretical Foundations
- 2) Secondly, impracticality of Testing All Data
- 3) Then, impracticality of Testing All Paths
- 4) Finally, absolute Proof of Correctness

VIII. CONCLUSION

Verification and Validation is a crucial part of the development life cycle of software. Verification and Validation both are necessary and complementary. Both of them provide their own sets of Error Filters. Each of them has its own way of detect out the errors left in the software. Verification starts from the requirements stage where design reviews and checklists are used to the validation stage. The cost and time of the project can be reduced by using verification and validation methods .

IX. REFERENCES:

- [1] Ian Somerville, "Software Engineering", Addison-Wesley.
- [2] Roger S. Pressman (2005), " Software engineering a practioner's approach 6/e"
- [3] Pankaj Jalote , "A Concise Introduction to Software Engineering", Springer 2008
- [4] Al-Neaimi, " Conducting verification and validation of amulti-Agent Systems", September 2012