# Iterative Validation Model/Star Validation Model for Improving Software Development Process

Sowmyalakshmi[1], Pooja Arora[2], Sakthi Kumaresh[3]

(II-BCA, M.O.P. Vaishnav College for Women) [1, 2]

((Department of BCA, M.O.P. Vaishnav College for Women) [3]

**Abstract:** Software process model is the general flow of functions carried out to develop a software. Currently there are many models used by different companies as per their requirements. Every model has its own features, benefits and limitations, but it is necessary to overcome the limitations for a become outcome and for the users' satisfaction. So, every new model produced has tried to resolve the limitations of the previous models thereby resulting in better software packages. In this paper, we discuss a new model where a new feature has been introduced, which is validation after every phase of development so that the changes needed are made before further development and the final product is one delivered with high levels of accuracy.

## 1. INTRODUCTION

One of the key aims of Software Development is to understand the customer's requirements in the software, and to make sure that the hinderances that brought about the need for the software in the first place are overcome. This is done by breaking down the process into various processes, starting with the documentation of the customer's needs as the client's requirements and ending with the final delivery of the product (which matches the initial requirements and functions satisfactorily). Any compilation of machine-readable code which functions to satisfy a specific need of the user is called software. This can include anything, from files to applications, basically anything that can be developed with the aid of computer code. With close analysis, though existing models have their own advantages, possess a number of traits which could prove to be useful if tweaked a little. This paper discusses modifications made on a new model which has primarily been derived from various existing models. The Iterative Validation Model, which is practically a user's model since it works to his advantage is flexible and reliable at the same time. These days, just like all quality things in life, the IV model too requires a price to be paid for its high levels of accuracy, time. It requires dedicated allotment of time from three people in particular, namely- The client, technician and the quality personnel. They say the good things come if you wait, and this is absolutely justified in the IV model since the final product is worth the wait and the effort it requires over the course of the development period. Since the model is validated after each of the phases that include requirement analysis, descriptive designing, construction, risk and defect analysis and, the product thus delivered is almost up to the level of software expected.

## 2. PROPOSED WORK

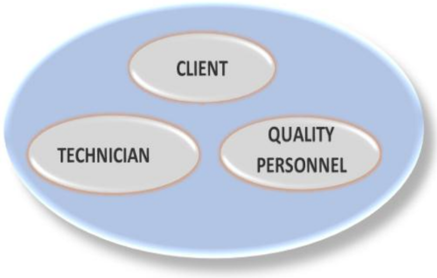
To develop a software there are some steps to be followed in an orderly fashion to deliver the final product as expected by the users. Every software development model has the same steps with some differences that make the particular model unique. With this paper, we present a model that will deliver the final product with high levels of accuracy as done by no model previously, as each phase of the model will be validated by the respective validators.

In every organisation, it is now very important for every person to agree with the software being developed and not every person's view is taken as development stages are reached.

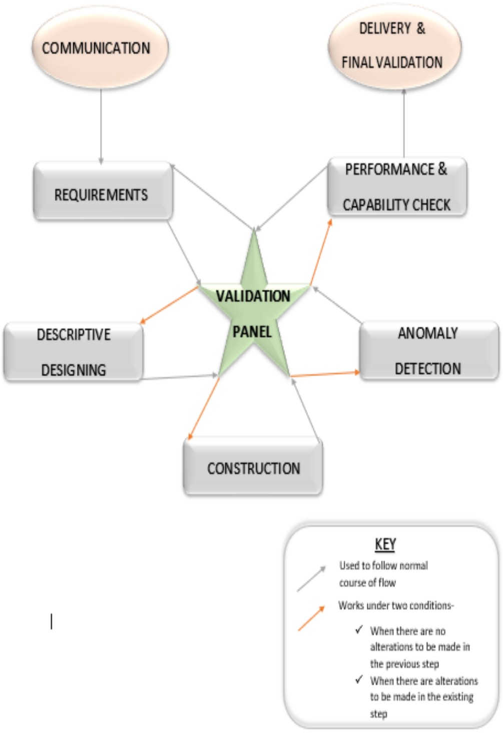
This model will serve to accomplish the task of taking the required personnel's views after every phase so the changes needed will be made immediately and not when the whole project is near its completion.

This model consists of 6 phases before delivery and at the end of each phase the sample will be presented before the validation panel (respective developers, users) for confirmation before the next phase initiates its processing. If used correctly, this model will result in the delivery of the final product with the kind of accuracy required and software expected by the end user. So, it will not only satisfy the customers but also the developers as maximum changes will be made on time. customers but also the developers as maximum changes will be made on time.

**(i)Diagram- Validation Panel Diagram**



**(ii)Diagram – Model Diagram**


COMMUNICATION
DELIVERY & FINAL VALIDATION
REQUIREMENTS
PERFORMANCE & CAPABILITY CHECK
VALIDATION PANEL
DESCRIPTIVE DESIGNING
ANOMALY DETECTION
CONSTRUCTION
KEY
Used to follow normal course of flow
Works under two conditions-
✓ When there are no alterations to be made in the previous step
✓ When there are alterations to be made in the existing step

## 3. VALIDATION PANEL PARTICIPANTS

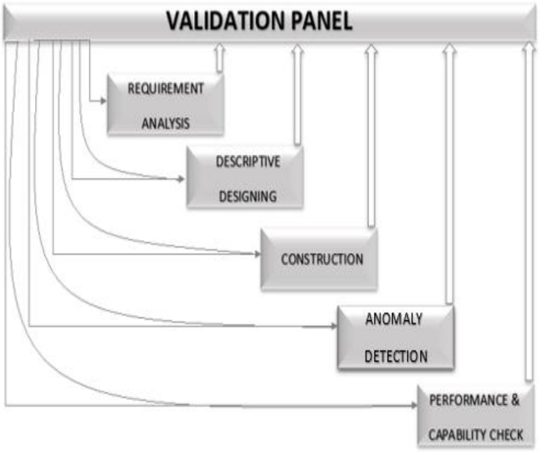
The main feature of the Star Validation model, is the validation panel which comprises of three main parties- The client/customer, the technician and quality personnel who isn't biased and provides an honest feedback about the proceedings of the development of the Software Model.

This figure represents the working of the entire model and each of its five independently functioning iterative loops. Starting with the communication between the developing team and the client, we proceed to the next step, or the beginning of the first iterative loop- Requirements phase, which deals with the conveying of the technical requirements to the client by the developing team. If this is acceptable to the user, and is validated by all three participants of the validation panel, we move on to the next phase which commences the next iterative loop. Else, the requirements phase is redone and sent to the validation panel over and over until when there are no objections or changes to be incorporated. The next phase or the beginning of the second iterative loop- Descriptive designing commences when the requirements phase has been done and the results has been finalized. In this iterative loop which again involves the validation panel, the design of the model which is proposed to be constructed in the next phase is presented for approval. If this is acceptable to the user, and is validated by all three participants of the validation panel, we move on to the next phase which commences the next iterative loop. Else, the descriptive designing phase is redone and sent to the validation panel over and over until when there are no objections or changes to be incorporated.

The next phase or the beginning of the third iterative loop- Construction commences when the descriptive designing phase has been done and the design has been finalized. In this iterative loop which again involves the validation panel, a basic prototype of the design is constructed for approval from the validation team. If the user feels that the prototype functions to his requirements, and is validated by all three participants of the validation panel, we move on to the next phase which commences the next iterative loop. Else, the construction phase is redone by applying modifications and sent to the validation panel over and over until when there are no objections or changes to be incorporated.

The next phase or the beginning of the fourth iterative loop- Anomaly Detection commences when the construction phase has been done successfully. In this iterative loop which again involves the validation panel, possible defects that might have not been identified in the previous stages and risks that might be involved in the functioning of the model are identified and brought to the notice of the validation panel. The validation panel is expected to go through the risks and defects and give their ideas and suggestions to eradicate the defects and reduce the risks until finally there are minimum or no defects and minimum risks. The next phase or the beginning of the fifth and final iterative loop- Performance and Capability Check commences after the Risk and Defect Analysis is completed successfully. In this iterative loop involving the validation panel, the software is made to run through various tests to check its capability and performance in performing its function efficiently. The results are presented to the validation panel till they are satisfied with its working.  The final step involves the final delivery of the final product to the client and getting back to them after a fixed period to get their final feedback after the usage of the software for that period.

**(iii)Phase Diagram**



The first phase commences after the communication with the customers, being the Requirement Analysis Phase where the requirements gathered from the customers using quality deployment function are analysed by the developers. After the completion of this phase the validation panel consisting of the users, technicians and the quality personnel validates the Software Requirement Document (SRS) produced by the developers. Once approved the next phase initiates and if not approved the same phase is repeated until approval is granted. The second phase is the Descriptive Designing Phase where the product is designed in detailed and the document popularly known as the Software Design Document (SDD) is presented. This document again is validated by the validation panel and the phase is repeated until approval. The third phase is the Construction Phase where the software is built. This is again sent for validation and the product is re-built until approved. The next phase is the Anomaly Detection Phase in which the product is tested for any defects that may not have been identified until then or could arise later when the software is actually put to use. Risk analysis is also done here to check for risks in using the model. The analysed document is then validated by the validation panel for approval. The fifth and last phase is the Performance and Capability Check Phase. This phase includes the checking of the software's performance and capability that is how accurate and reliable the software is as compared to its expectation. This is validated and once approved by the validation panel the product will be qualified to be delivered to the customer.

## 4. WORKING

**Communication-** The process starts with the interaction between the customer and the developer, taking into consideration the requirements of the software along with proper documentation.

**Requirement Analysis/Elicitation-** Quality Function Deployment is the type of Requirement Elicitation being used for the Iterative Validation Model, thus three requirement types are identified from the client's needs namely, Normal, Expected and Exciting requirements

Then, the developer has an elaborate discussion with the working team and comes up with the requirements essential in order to develop the software with the necessary characteristics. After this, the requirements of the development team are presented to the validation team for approval.

In case there are alterations to be made, the requirements process is redone until there are no objections from the panellists.

**Descriptive Designing -** This is when the actual process of initial drafting of the design of the software takes place.

After this phase, the design is sent across to the validation panel for approval. In this step, mainly the technician's role is extremely crucial as he knows the technical nuances & purpose that the model needs to serve.

In case there are alterations to be made, the designing process is redone until there are no objections from the panellists.

More time is spent in this step as it is crucial to try to get a reasonably successful design in order to build a successful model, while saving resources.

**Construction -** The design is then implemented into a real time sample to be presented to the validation panel for a thorough trial.

In case there are alterations to be made, the existing model is updated by going through the building process again until there are no objections from the panellists.

**Anomaly Detection -** Once the previous step receives validation from the panellists, the model goes through risk and defect analysis to identify probable risks and hinderances/problems in its working, including technical issues like cost underestimation, schedule delays, or overrun of the resources of the project.

In case there are alterations to be made, the defect analysis process is repeated until there are no objections from the panellists.

**Performance and Capability Check-** This is the final stage of the software's development before actual delivery is testing. During this phase, the final software is verified to ensure that the user's requirements are met.

To double check, this is run through the panellists. In case there are alterations to be made, the model goes though this process until there are no objections from their side.

**Delivery-** The delivery process in this model is quite unique as the user receives a set review period for which they get to use the software and confirm its accuracy.

## 5. CHARACTERISTIC TRAITS OF THE MODEL'S EFFICIENCY

1. Reliability
2. Accuracy
3. Transparency
4. Quality
5. Flexibility
6. Resource management

1. Accuracy –
   The accuracy of the final end product produced using this model is higher because of the constant validation being received at the end of each step. Also from the user's point of view, the constant interaction with the developers and technicians added to the constant updates they receive help in achieving this level of accuracy.

2. Reliability-
   The reliability of the software packages developed using this model is higher than those which used other models. This is directly related to the accuracy because when the accuracy of the product is higher, so will the reliability. The user constantly interacts with the developing team, conveying his requirements at the end of every stage, making the end product more reliable.
3. Transparency-
   Since the amount of interaction between the user and the developing team is high, everything is transparent and the user is always informed about the status of the development and has clarity at any point.
4. Quality-
   The quality of the software, thanks to the provision of acquiring validation after the end of each step, and the quality personnel involved in the same. The quality personnel is basically a third-party person.
5. Flexibility-
   The validation panel which consists of the user, technicians and the quality personnel get the opportunity to constantly offer suggestions and modifications to the software model, that is the model is flexible and is constantly undergoing changes throughout the course of its development. Even though extra cost is incurred due to flexibility, the overall cost is saved as resources consumed in the development process of the software are minimum.
6. Resource management-
   Unlike other process models which generally involve discarding of sample versions of the software once there are changes to be incorporated in them, this model subjects one single version to all the alterations. Also, in this process model, the probability of facing criticism once the entire software is built is very low as the problems with it are rectified as and when they are identified at the end of each step.

**(i)Table COMPARISION BETWEEN EXISTING MODELS (RAD, SPIRAL, AGILE) AND ITERATIVE VALIDATION MODEL**

| ***ATTRIBUTES*** | ***RAD MODEL*** | ***SPIRAL MODEL*** | ***AGILE MODEL*** | ***ITERATIVE VALIDATION*** |
|---|---|---|---|---|
| *ACCURACY* | The prototype is delivered rapidly so the model does not produce very accurate products. | Accuracy is achieved only after multiple prototypes are constructed. | The product is validated after every iteration to achieve accuracy, but due to the many iterations a lot of time is consumed in achieving the same. | The product is validated after each phase so the final delivered is highly accurate. |
| *RELIABILITY* | As accuracy is directly related to reliability and this model does not produce very accurate products it cannot be relieved upon for expected results. | The spiral model can be relieved upon to an extent as multiple protypes are constructed so, the final product will be almost as expected. | The product to be delivered is reliable but the time taken to achieve the reliability is higher than in any other model due to the many iterations. | Accuracy is high, so is reliability as the concerned party is always informed about the changes made to the model. The final product delivered will be as expected. |
| *TRANSPARENCY* | Involvement of the customer is the key part of the model [5]. | Designers and programmers actively involved in the review process after each phase [1]. | Customers and stakeholders review the product after every iteration. | Involvement of the customer, quality personnel and technician is crucial to this model's success. |

| | | | | |
|---|---|---|---|---|
| *FLEXIBILITY* | The prototype is used by the users and necessary changes are communicated to the developers [5]. | Changes are incorporated in the consecutive prototypes.<br>After the end of each cycle of the four phases [3]. | Flexibility is high as the product is validated after every iteration but additional cost to make the changes is vast as each iteration is a time-consuming process [4]. | The documents prepared are given to the users and other personnel for approval and changes to be made are discussed by the panel together, before the actual delivery of the software. |
| *RESOURCE MANAGEMENT* | A working prototype is given to the user after every phase for validation. | A new prototype is developed at the end of each cycle of the four phases so, resource management isn't desirable in this model. | Resources are not saved as changes are made after every iteration consisting of five phases and repeating each iteration consumes a lot resources [4]. | A number of validation tests are done before the actual construction of the software, because of which no resources are actually wasted in the construction of the model. |
| *PRESENTATION TO THE USER* | A working sample is presented to the user. | The software is not presented to the user until the end of the last phase [6]. | The model is presented to the user after each iteration consisting of five different phases. | Starting from the initial documents everything is presented for approval. |
| *PATTERN* | It is a sequential model comprising of consecutive steps. | It is an iterative loop of all the four phases combined. | It is an iterative loop consisting of five phases combined [4]. | It comprises of five iterative loops, each of which function individually with the validation panel. |

## 6. PROJECT STUDY

Two projects developed using traditional process model were taken for study and their relative metrics were calculated using Cost Constructive Model (COCOMO). Table 2 shows the various metric values for projects developed using traditional model

**Table 2: Calculation of various metric values for two projects using traditional process model**

| | KLOC | E | D | P | UVC | Verity |
|---|---|---|---|---|---|---|
| **Project-1 (Graphics game)** | **2.37** | **5.937** | **4.917** | **0.399192** | **1** | **0.399192** |
| **Project-2 ( Java game)** | **2.289** | **5.724** | **4.85** | **0.399895** | **2** | **0.79979** |

The effort is calculated in using the formula:

EFFORT (E) $=a_b$ (KLOC) $^{b_b}$ [5]

DEVELOPMENT (D) $=c_b$ (E) $^{d_b}$

Where $a_b$, $b_b$, $c_b$ and $d_b$ are co-efficients and are tabulated with categories organic, semi-detached and embedded.
The values for the four coefficients are taken as 2.4, 1.05, 2.5 and 0.38 respectively.

The productivity of the above projects is calculated using the formula:

Productivity (P) = KLOC/E [7]

User Validation Count (UVC) is nothing but the count of the number of times the user's validation is obtained through the course of the construction of the software.

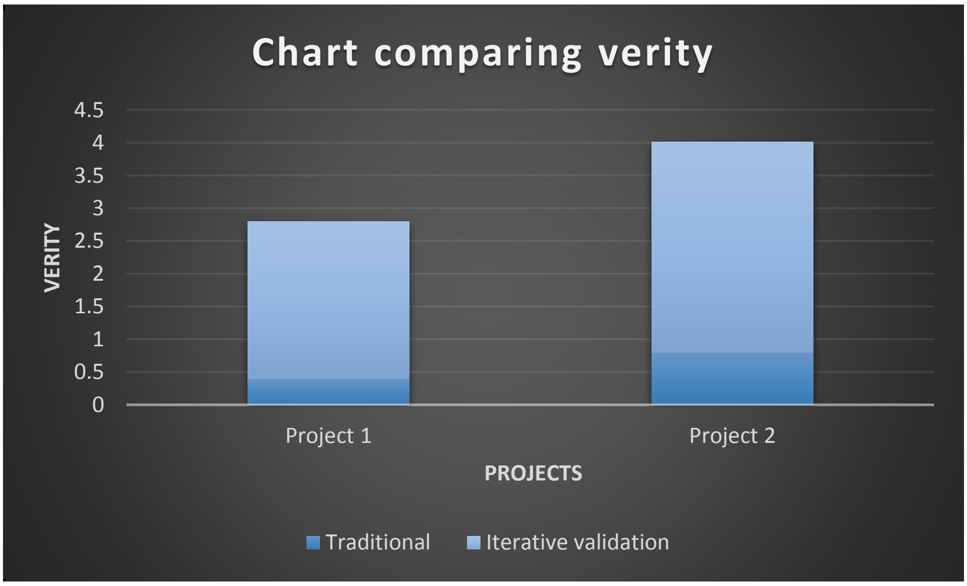
As defined in [10], "Verity is the quality or condition of being true, factual, or real." This is calculated as follows:

Verity = P*UVC

**Table 3: Calculation of various metric values for projects using Iterative Validation Model**

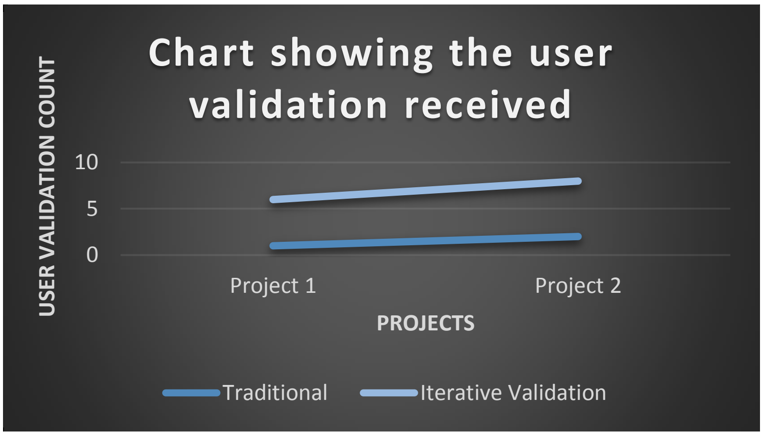
| PROJECTS | KLOC | E | D | P | UVC | Verity |
|---|---|---|---|---|---|---|
| **Project-1 (Graphics game)** | **2.21** | **5.517** | **4.782** | **0.40058** | **6** | **2.40348** |
| **Project-2 (Java game)** | **2.104** | **5.239** | **4.69** | **0.401603** | **8** | **3.212827** |

The above data shows the projects that are implemented using the iterative validation model. It is noted that the User Validation Count and the verity is more when compared to the projects that are developed using other basic and traditional models namely, Waterfall model and prototyping model which is graphically depicted in diagrams (iv) and (v).



**(iv)Diagram - Chart showing comparison of verity**

The graph shows the quantitative difference in verity, of the two projects developed, using traditional software development models and iterative validation model. As verity is derived arithmetically by the product of productivity (P) and user validation count (UVC), it is plotted numerically on the graph as the y-axis. The two projects are shown along with each of the models used, as the x-axis. The difference in verity is clearly shown as the two differently shaded regions, which implies that verity is higher when the projects are developed using the Iterative Validation Model than the verity derived when they are developed using the traditional models.



**(v)Diagram – User Validation Received**

The above graph represents the amount of user validation steps involved throughout the course of the development of the software. As we can infer, the user validation is higher in the Iterative model when compared to the traditional models.

## 7. CONCLUSION

The table containing data and the graphs clearly show that the number of user validation and verity of the software is considerably higher when projects are developed using the iterative validation model.

The advantages clearly state about the accuracy and reliability of the model along with its flexibility as the changes can be made immediately as and when required. The limitations are also clearly stated as the model is a time-consuming process model due to the validation after each phase for the respective changes necessary. Also, as all the members of the validation panel may not be available at the time when required this model has its drawbacks.

While the iterative validation model can also be used on small projects such as the one tested, it is most suitable for larger projects where customers are actively wanting to be involved after every step and the technicians and quality personnel are available at the necessary times required. The metrics may differ from smaller projects to larger projects so the iterative validation model can be selected based on the kind of people involved during the development of a software.

## REFERENCES

[1]https://www.researchgate.net/publication/280600317_The_Evolution_of_Software_Process_Models_From_the_Waterfall_Model_to_the_Unified_Modelling_Language_UML

[2] http://www.ijsrp.org/research-paper-1013/ijsrp-p2242.pdf

[3]https://www.researchgate.net/publication/298656663_A_survey_of_software_development_process_models_in_software_engineering

[4]https://pdfs.semanticscholar.org/ed4d/0cf18b05cb9bc7ef3bc9e8d51f58d722486f.pdf

[5]https://www.tutorialride.com/software-testing/software-development-process-models.html

[6]http://www.niecdelhi.ac.in/uploads/Notes/btech/4sem/cse/21378403-Software-Engineering-By-K-K-Aggarwal-YogeshSingh-Full-Notes.pdf

[7]http://www.ijsrp.org/research-paper-0313/ijsrp-p1590.pdf

[8]https://en.wikipedia.org/wiki/Software

[9]https://www.geeksforgeeks.org/software-engineering-halsteads-software-metrics/

[10]https://www.thefreedictionary.com/verity