Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 1804–1814

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Optimization of Software Testing

Mundita AWOTAR[a], Roopesh Kevin SUNGKUR[b]

[a,b] *Department of Software and Information Systems, Faculty of Information Communication and Digital Technologies, University of Mauritius, Reduit, Mauritius*

Abstract

The goal of any business is to satisfy the needs of its target customers, and IT industry is not an exception from that rule. Thus, the upgraded version of the V-model testing is supposed to deal with the weaknesses of the original version in question by combining it with the method known as agile testing. At the beginning of the report, hypothesis such as the strengths and weaknesses of the existing V-model testing via literature review and interviews with respective specialists in the sphere were analysed. Successively, the possible advantages of agile method of testing were then considered. Moreover, the report comes up with the ways in which the two models could be naturally combined to produce a much more effective one. Once the new model was presented, its strengths and weaknesses were assessed by the means of a case study analysis using metric and a data analysis through a survey were conducted to evaluate the credibility of the futurist model. Promptly, the research found that the suggested testing model provides better results than the common version of V-model testing. Firstly, a real case scenarios under metric evaluation of the models have indicated that the proposed model is better than the V-model, since it can handle the following aspects; reduced testing time, debugging, prioritization of requirements , easy mapping of roles and improved visibility of project resources. Secondly, a survey data analysis highlighted various advantages of the future model. The top priorities of the new model from the respondent's perception were; the new model manages rapidly changing priorities, it accelerates time to market, it increases productivity and it improves quality.

© 2018 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/3.0/)
Peer-review under responsibility of the scientific committee of the International Conference on Computational Intelligence and Data Science (ICCIDS 2018).

Corresponding author: r.sungkur@uom.ac.mu

1877-0509 © 2018 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/3.0/)
Peer-review under responsibility of the scientific committee of the International Conference on Computational Intelligence and Data Science (ICCIDS 2018).
10.1016/j.procs.2018.05.142

*Keywords:*Software Testing; V-model, Agile Development; Requirements; Debugging

1. Introduction

With There are a myriad of Software development processes and methods available that can be used in the development of software, some of which are traditional and are termed as "old-fashioned" by proponents of newer and more "classy" processes and methodologies such as Agile methodologies. The more traditional approaches to software development include the waterfall and V-model methods which have been in use in software development cycles for decades, but are regularly the point of critics. The focus of this paper will primarily be on the V-model of software development combined with the testing process of more modern approach of software development known as agile development. The V-model of software development is regarded as an extension of the waterfall model and in this type of methodology; software development execution of processes happens in a sequential approach with a V shape that involves a sequence of processes and is also regarded as the verification and validation model. The V model is regarded as a high level design of Test Driven Development (TDD), and each development phase of the software is directly associated with a corresponding testing phase. This means that each corresponding phase of testing is planned in parallel with the development phase hence test cases are developed in the development phase in order to be implemented in the corresponding testing phase, but typically testing is conducted once the software is completed. As mentioned above it consists of two aspects, verification and validation. Verification is an internal process as compared to validation and typically involves checking if the software complies with its technical specifications as defined and guided by a system analyst. Validation, on the other hand, involves the compliance of the software to the requirements, needs or specification of the customer. In this case, the left side of the V model describes the various business and technical details while the right side is more concerned about the testing.

Software Testing is a critical task which is meant to ensure that the software developed meets the requirements specified by the user. It is a demonstration of quality of the software to its stakeholders hence software testing is an integral part of Software Development. Agile testing is a software technique that borrows heavily and directly pursues the rules of Agile Manifesto where software development iteratively and incrementally as it actively involves the stakeholders. Agile development involves the iterative and incremental release of software hence the need to test software frequently. Agile testing comprises every member of a cross-functional team but requires the expertise of a specialized tester. This study attempts to improve the existing V-model testing and to decrease the high level of uncertainty and risk which were encountered by both testers and developers on the final stage of product development. Flexibility is the next major aspect that has been catered in the futurist model. Moreover, end users would get fully operational products of high quality in minimum cost as compared to the traditional V-model. Companies that shall adopt the modified and improved model will benefit from the resources on getting rid of the bugs, unnoticed by both developers and testers. Besides, such an approach would also be beneficial to businesses, since their expenditures and issues related with lateness would be minimized at least, if not eliminated completely. As clearly mentioned above, adopting agile development testing in V-model is a wise approach to tackling the problem of unpredictable system needs.

2. Literature Review

A. Traditional V-model Software Development

The traditional V-model Software Development Life Cycle is a Test Driven Development (TDD) software development method which involves defining, developing and evaluating test cases prior to writing the actual code. This involves real planning in anticipation of what the development environment will be. After the project scope is defined and planning is done before the verification and validation phases, tests are consequently developed for the

test phases. In the case of "wicked" projects where the problems are not clearly defined, the scope of the project is vague or the specifications submitted by the user regularly change where the stakeholders of the project have an idea of what they require, but it is unclear and they believe that the picture will come out once development commences and a prototype is subsequently released. In this case, it becomes difficult for test cases to be developed by the business analyst and the software validation plan cannot be written down.

This brings out the following limitations of the V-model:

- Rigidity and inflexibility: The inability to respond to changing requirements and be able to adjust development.
- The concentration of software development into one phase (implementation phase).

B. V-model

The V-model is also referred to as the Vee-model, which is a software development process. According to Rouse (2007), the v-model origin is Germany where it was used in the development of products for its government's defense projects. According to Firesmith (2013) is a creation from the changes revolving around the waterfall development model. The flowchart of the V-model is v-shaped hence its name. The flow of the development process starts from the left upper side of the V to the right side. Firesmith (2013) states that verification and validation is the most important part of software development especially the testing process. Requirements verification and validation are the two most critical parts in software and system development. According to this writer, software verification and validation were the main strategies behind the development of the waterfall model, from which the V-model is generated. In another aspect, V-model development process is on an association basis of the testing phase for each of the corresponding development stages. This when put in other words means that there is a direct association of each development phase with a testing phase. Each step starts after the full completion of the preceding phase.

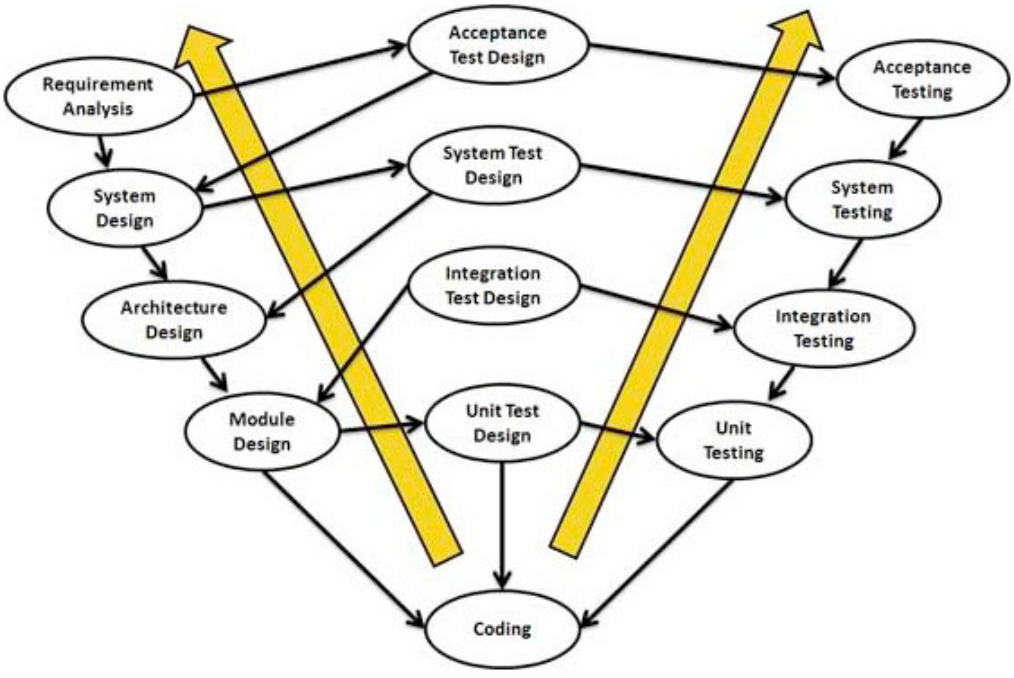

Figure 1: V Model

The image above shows a better representation of the V-model development process, where the left side of the V represents the verification process and the right side represents the validation process co-joined by the codding phase. Each development phase has an alternate testing phase on the other side of the V shape (Tutorialspoint.com, 2014). The representation of this model is that the phases are sequential rather than the usual iterative, incremental or concurrent phases in agile development processes. V-model presents a simplification of the development process as a whole. Like any other development model, the V-model has its characteristics, advantages and disadvantages. For a developer to use the V-model for software development, the developer require to have well-defined fixed and documented requirements, be working on a short project, the technology to be applied is not dynamic and the product definition should be stable. Clarity of the requirement is necessary because going back to correct a mistake can be very expensive. Flexibility is main advantage (Bucanac, 1999). According to Regulwar et al (2010), the verification and validation model of development is commonly known as the V model and is an extension of the common waterfall model as considered by many. Its similarity with the waterfall model is its sequential phase progress and linear development. Regulwar et al (2010) agrees that the initiation of any phase in the process is after the successful completion of the preceding phase. These writers note that testing activities in this model all occur before the coding process for instance the planning and test designing. Regulwar et al (2010) states that this is an important aspect of the V model development process. Due to the early testing activities, the testing team is involved early in the development process which save time and help the team gain better understanding of the product from its initial development phases. The early testing is also advantageous in that the identification of faults is early in the development process and it is not expensive to rectify them. The problem with this model according to the writers is that the model focuses on the dynamic development phases instead of the product itself.

C. Agile Testing

Agility involves a number of variations. It includes the effective, adaptive and rapid response to change, it includes the effective communication among the stakeholders, it involves the inclusion of the customer onto the development team and it involves the team organization in a way that it can control the work performed (Turk, 2005). In this case, the writer establishes a way to deliver and agile software. The process is driven by the descriptions of the customer according to the scenarios they require, the process recognizes the short-lividness of the plans, the development of the software is iterative paying heavy emphasis on the activities of construction and the process has multiple software increments delivered. According to Turk (2005), agile software development is about a close relationship between the developer and the customers during the development process so as to understand the requirements of the customer. This aids in the development of a product that is closely in line with the business requirements and the delivery of the end product is much faster. In regards to the writer's paper, developers that practice agile software development encourage change in the early phases of the project development process rather than discourage them at the final development phases when deploying the software. According to the writer, there is stress on the quality of the design as well as the requirements testing, constantly in the early stages which is less expensive and easy to detect. Below is a comparison between agile and traditional software development such as the waterfall model.

Agile software development requires the presence of the client at all time and it is hence termed as face-to-face. According Pelrine (2011), the companies and developers that choose to use agile software development have more success when compared to other development methods. According to Verwijs (2012), there are five principles encompassed in agile software development that improve the general agility of the process. This principles include; 1) Just in time coding and design which is about efficiency and economic distribution of time, 2) think, write, test and refactor since the development process is like solving complex problems and it can be a tricky process, 3) unit testing 4) deal with object oriented code instead of procedural codes and 5) application of agile design principles and patterns such as dependency injection ad single responsibility principle among others. According to Misra (2007) agile software development is a result of the agile philosophy formulated in an effort to combat all other development models challenges. All the other development model were realized by many different software practitioners and the same people criticized these models. The agile philosophy was realized in an effort to solve the challenges by seventeen software practitioners who teamed up. This group developed the Manifesto for Agile

Software Development which provides the values and principles of agile software development. The two images below will provide the values and principle of the philosophy.

Adopting agile development in V-model

The first step in this case requires the comparison of the advantages and disadvantages of the two as mentioned in the above two sections. This allows the researcher to identify the differences between the two which are represented below.

Table 1: Differences between Agile and V-Model

Agile software Development	V-Model software development
Works to deliver a working product at each end of iteration	Does not have iterations
Contains every phase of software development	It does not have all the phases of software development such as maintenance, repair, operation and disposal
The developer, tester and customer work together on each development stage	Developer works on designing, coding and testing. There is no working together.
Works with project whose requirements change often	Requires clarity of requirements before initiating the development process.

D. Critisism

The V model is about saving time and assuring that the product is corrected at each stage though the testing from the early development stages. It takes that each phase must be complete before the other one begins. Agile development is beneficial in that it incorporates the customer at each stage of development from the initial stage of requirement definition. It is also about incremental delivery of the software. The adoption of agile development into the V model simplifies the wait for the late delivery of any finished product. The process also eliminates the high probability that the developers do not understand the requirements by the inclusion of the customer through the development process. Complexity of the software is eliminated through the collaboration of the development team and the customers. This adoption does not come without its limiting factors. The process is about breaking down the whole sequential process into smaller sequential processes with shorter time. This might save time which increasingly consumed during the coordination process among the agile teams. This might end up increasing the general time used. Again, if the first requirement is not understood, the whole incremental process goes wrong. The adoption of agile development into the V model has its advantages and disadvantages for sure.

3. Methodology and Proposed Solution

The study has been conducted within the frames of design research when an upgraded version of the V-model testing is supposed to deal with the weaknesses of the original version in question by combining it with the method known as agile testing. The research consisted of the following:

- Highlight the possible advantages of agile method of testing.
- Analyze the ways in which the two models, traditional V model and the proposed one could be naturally combined to produce a much more effective one
- Design the new framework
- Evaluation of the new model in real life scenario using Metrics and analysis of survey's data collection.
- Assess the strengths and weaknesses of the new model

In order to test that hypothesis such methods as literature review, interviews, surveys and case study analysis were

used during the study. The research found that the suggested testing model provides better results than the common version of V-model testing. Qualitative method such as one-on-one interview was used to discover key topics such as, the advantages and disadvantages of using traditional V model in real companies among others. Software engineers, Teams Leads, Trainees and Managers were interviewed to study the challenges they faced during using the V model Framework in their daily operations. Moreover, researches has been done through the internet source, magazines and academic articles on the 'new methodology adopting agile development testing in V-model' to improve different aspects such as Time, Resource, Quality and Cost of a company. Once the new framework with agile development was proposed, a case study was used to evaluate its credibility. Real case scenarios helped to learn the behaviours of both models, the traditional V model and the proposed new model adopting agile development testing.
However metrics and analysis of data collected from a survey were used to measure the new proposed model in terms of benefits that the new framework will provide.

4. Designing the New Framework

As clearly stated in the problem statement above, to solve the problem of unclear or changing requirements, the proposed methodology is the implementation of agile development testing in V-model. Agile methodologies are many and vary in design and implementation depending on the project and purpose. In this particular situation, the agile methodology of choice is Scrum. The diagram below shows the traditional V-model architecture.

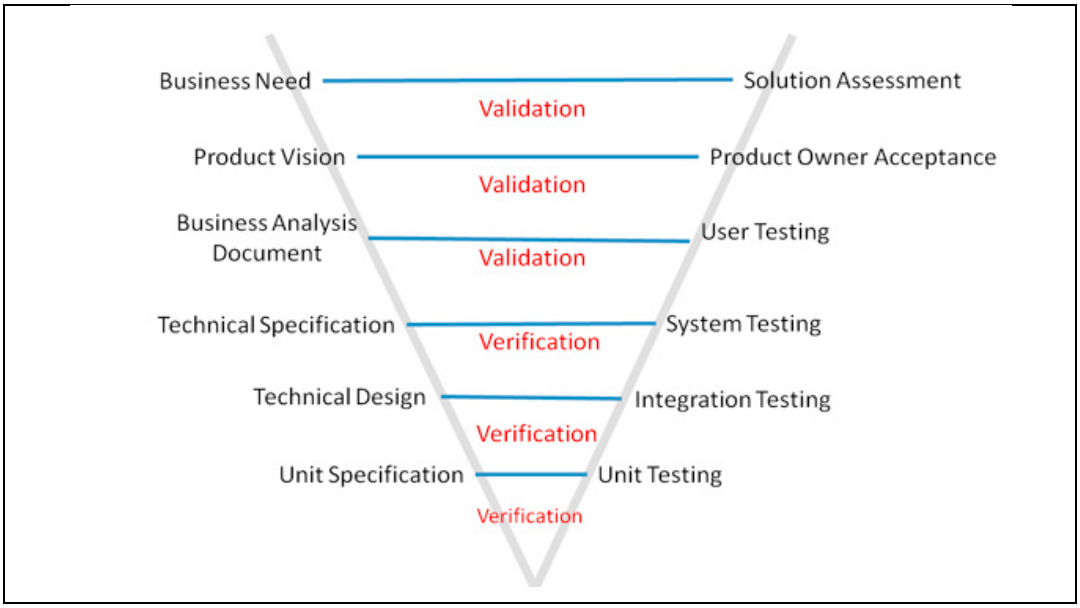

Figure 2: Simplified V Model for Traditional Development Testing

A. Proposed Model

The suggested agile V-Model will have four testing levels and will also have more proper documentation. This proposal combines the aspects of a V-Model and an agile model (Monteleone 2014, n.p). V-model is a software model that entails building a V-shape sequence for testing techniques involved with a specific design. Testing and coding involved in the V-model contributes to the process of software development (Venantius Laulin 2014, n.p). The proposed Agile V-model incorporates the features of agile model and V-model. Below is the proposed model. In the proposed model, of agile development testing in V-model, the sequential implantation of phases becomes slightly different whereby four phases of testing has been implemented.

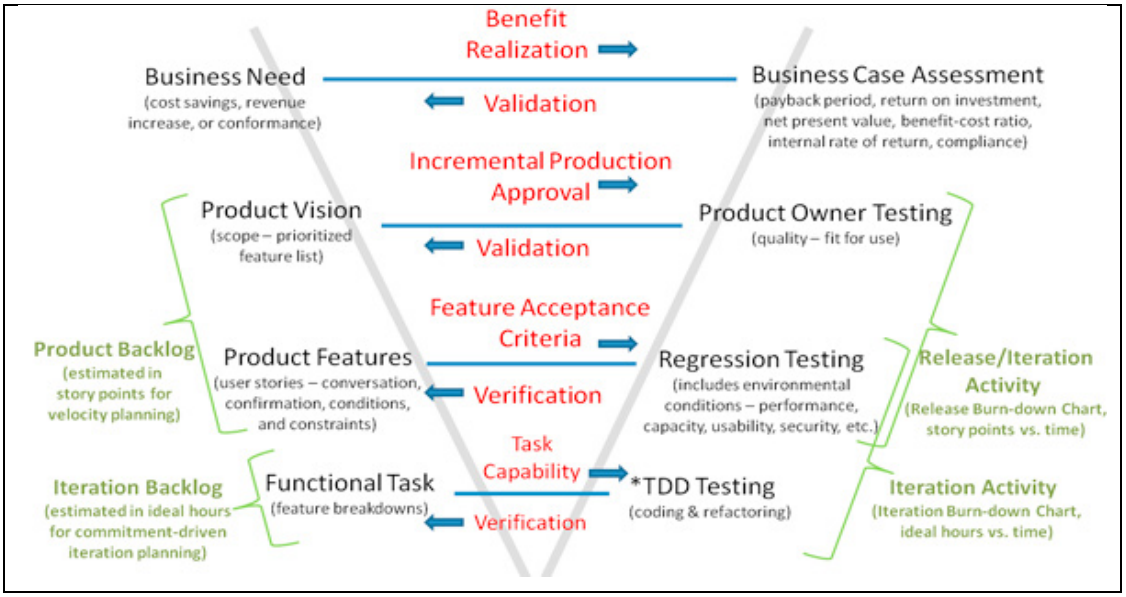

Figure 3: Proposed Agile V-model

B. Mapping the processes to the characteristics

The features of the V-model remain as seen on the left part of the diagram. There is a development progress to come up with solution deliverables of high quality. The first sequence is the business need. It is then possible to create solution given that from the model one can develop software. From the business need, one looks at the product vision, product features and the functional tasks (Beal 2014, n.p). Agile model comes in by breaking down these features and tasks through several steps. It critically looks at all areas of the features involved in the V-model by calculating, inputting, transmitting and printing all information to help in correctly estimating the outcome of the features and whether it presents the preferred solution. These steps include validating benefit realization; validating product approval, verifying acceptance of the features and also verification of capability of a functional task. The right-side of an outmoded V-model indicates the progression and development of the incorporated testing levels that were in the middle. For example, in this case the first sequence on the left side was identification of the business needs of the company. On the right side, a case assessment may be done using economic indicators to ensure that the benefits of the company will be realized. These economic indicators include calculating investment return, payback period, net-present value and benefit-cost ratio. For the product vision, the owner tests the product for its quality. Product features are tested through regression analysis to ensure capability, performance and usability. Functional tasks undergo a test-driven development analysis to refactor and code (Ambler 2007, n.p).

5. Results and Discussions

This section presents a case scenario of a software project and also presents a comparison of suitable development Software analysis methods. A case scenario of a health care system development was conducted to highlight the distinct aspects of the two software-testing approaches and the advantages that the proposed model, software development approach, Scrum has over the V-model in software testing. Scrum is a software development approach that is founded on the principles and concepts of agile software development, it supports both incremental and iterative development. On the other hand, the V-model is traditional approach to software development that is characterized by upward progression of phases, after the coding phases, to form a characteristic V shape. This proposed model shows the relationships that exist between each development life cycle (DLC) phase and the allied testing phase.

A. The Case Scenario

Hospital XYZ has been developing a proprietary healthcare management system internally. The system is large, customized and highly complex with four core modules and six support modules. The core modules include the patient and care management, electronic medical record and departmental modules. The support modules include the billing systems, dashboards, housekeeping, HR management, accounts and financials, and inventory management. The organization is a large commercial healthcare provider operating in a highly competitive healthcare industry. In the near future, two other branches which will be located in two different states of the United States are already in the pipe lines. This multisite aspect of the organization poses a significant challenge to the implementation and testing of the healthcare system. Being a pioneer project for the enterprise, issues such as unclearly defined scope and vague requirements characterized the project. Novelty, proficiency and wide experience in computer system design and development as well as in health care provision and management are factors that have significant implication on the success of the project. The management of Hospital XYZ expected the project to be completed with a year. The V-model can be used as the primary methodology for the designing, developing, implementing and testing Hospital XYZ's healthcare management system. The methodology connects early development tasks (or activities) to the corresponding testing activities conducted later in the development cycle. The development of Hospital XYZ's healthcare management system has been done under the V-model. However, the proposed model presents a suitable methodology for the design, development, and implementation and testing of Hospital XYZ's healthcare management system, where the testing tools are linked to the collection of requirements and design and which are reflected in a descending order to the application of verification and validation tools. Equal team resources is given to coding and testing within the model, thereby requiring that coding be carried out at the same time as the testing and documentation of the various models that are developed in a single sprint. The idea here is to have an implementation plan that precludes a testing and documentation phase for each single sprint of a development component in order to reduce the occurrence of discrepancies within the development process rather than waiting for the completion of the entire system so as to handle the testing and documentation.

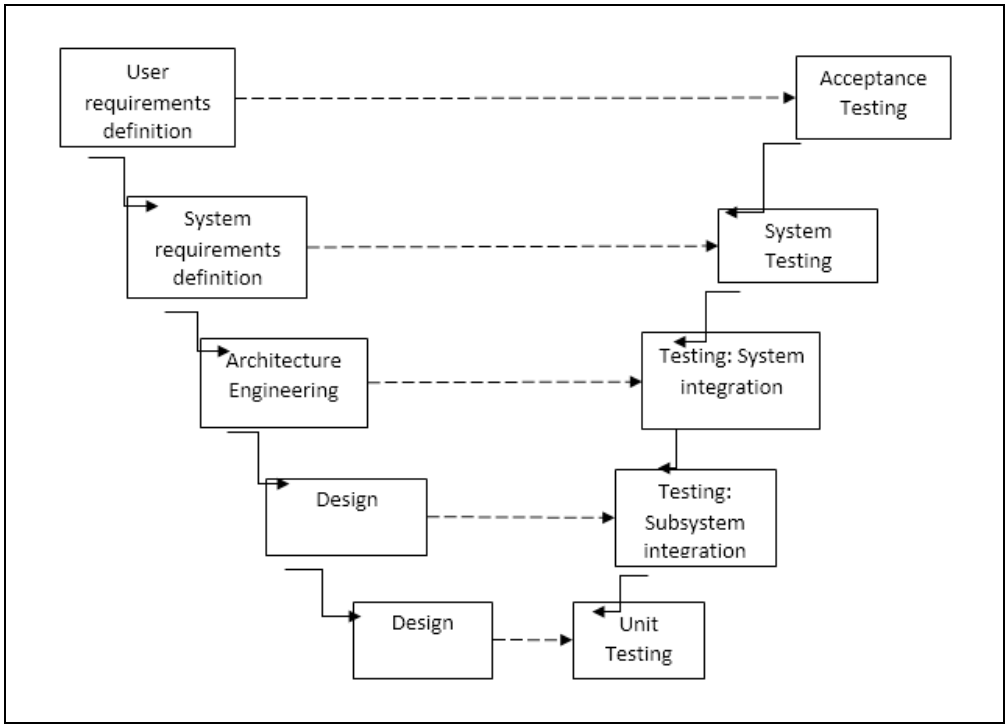

Figure 4: V Model for Case Scenario

B. Case Study Evalutation and Discussions

Company A situated in Mauritius get the project for the Hospital XYZ. One of the authors was granted the opportunity by Company A to conduct data analysis for the software to be developed for the Hospital XYZ with the Mauritian team's members on this report purpose. Moreover one to one interviews was organized by one of the authors with some key team members for this report and eventually, the results that were gathered, highlights the benefits of the modern (and futuristic) software development approach, Scrum, against the traditional V-model using the real life scenario, Hospital XYZ.

The proposed model, agile development in v-model, has potential benefits over the traditional v-model as listed below:

- **Reliability** – The New model is much more simple and easier to use than the traditional V- model thus more suitable for software analysis.
- **Less costly** – The use of methods such as the Internal Rate of Return (IRR), Net Present value (NPR) and Return on Investment (ROI) presents a structure that minimizes the cost of design and production as it carefully evaluates system requirements before the implementation phase.
- **Reduced use of resources and time** – The proposed model presents a much more compressed and streamlined analysis system that allows for proper time keeping strategy.
- **Less defect fixing** – This system has been designed in a way that avoids the downward flow of the defects in its flow.
- In addition to the benefits, the proposed model is better than the v-model in terms of **handling the risks** associated with the particular health care information system project highlighted by the team's members.

C. Validation Results through Metrics

This section is where the derived metrics have been applied to the case scenario of Hospital XYZ. Both methodologies have been monitored. For the first case scenario, the application of V model methodology in proceeding with the development of the product and on the second hand, assumed but merely justified facts of key members have been applied in the proposed model. Below of the results of the evaluation done for the system of Hospital XYZ. By doing the comparison of the traditional V model against the proposed Model, the following aspects have been illustrated in the favors of the futuristic model.

- **Reduced Testing Time**

The proposed model works by the integration of the varied capabilities of the agile development methodology and the traditional V-modeling solution, reducing the time which is spent during the software development process by carrying out testing of each module at end of a single development sprint. This presents the advantage of reducing the overall testing and debugging time as the testing is carried out in parallel with coding and development process. Additionally, the use of the iterative nature of the agile development methodology implies that the system features are delivered in an incremental basis, thereby allowing the realization of some system benefits at the early phases of the system development. This will increase stakeholder confidence and generation of requirements, which will effectively lead to the development of a system that duly serves the required operations. The key principle of the proposed system is that the testing and possible debugging of the system is carried out throughout the entire lifecycle of the system development thereby enabling regular inspection of the deliverables at each sprint. This will enable the stakeholders to make the necessary adjustments (where necessary) and give the development team early signals of any operational quality issues that require to be corrected. From the development case study, the V-Model took three straight weeks to test the working compatibilities of the system in relation to the requirements of the stakeholders. This is related to the fact that V-model testing involves the analysis of the entire system in relation to the user needs, thereby taking a comparatively longer time to determine the applicability of the met user requirements. The proposed integrated model took six separate days to test each single development sprint. This is because the software developed was composed of six different sprints, each with a re-evaluation of the stakeholder requirements. For each sprint deliverable, agile testing was carried out thereby effectively reducing the bulk of the

system development testing time. Each testing session lasted a single day since previous testing sessions had eliminated any operational and requirements inconsistencies within the previous deliverables, significantly reducing the amount of time required for testing.

- **Debugging**

The proposed model is designed to carry out iterative debugging at the end of each single development sprint. This implies that the software development team is required to carry out subjective debugging of the developed module within each sprint, removing any developmental and operational bugs that might hinder the collective operation of the integrated system. From a developer's point of view, this reduces the effective total time that is required to debug the entire system, together with the load that is subjected to the developers and the ease with which they can be able to find the associated bugs. Accordingly the case study revealed the following debugging time measures:
Debugging using the V-model took four weeks to accomplish, since it necessitated the collective debugging of the entire system so as to ensure that all the system modules worked as required. Comparatively, the proposed model took 9 separate days to debug each of the six sprint instances (each of the debugging sessions taking approximately 1.5 days). With the presentation of this debugging in phased model, the stakeholders were able to point out a number of system operational inconsistencies that were easily corrected in the following sprints.

- **Prioritization of requirements**

By means of system deliverable backlog and prioritization of user requirements for each single sprint, the proposed system will allow the system developers to undertake riskier and valued requirements during the initial stages of the development cycle. Accordingly, this development procedure allows the development of certain procedures and functionalities within the early sprints as compared to other non-prioritized functionalities. In this regards, the proposed system made it possible for the development of the login and records sharing interfaces during the early sprints, which were delivered to the stakeholders for evaluation. This positively influenced the generation and development of user requirements in a rapid manner, since the users could be able to envision what the system could do and elemental functionalities that they wanted to add to the system to make it better. Sharing of records and assignment of operational responsibilities was achieved early on during the development, thereby allowing the development of related interfaces in accordance with the set requirements.

- **Improved visibility of project resources**

The proposed model will allow for increased visibility by means of tasks and test boards together with daily/weekly sprints. Accordingly, the proposed model is designed to increase the visibility of the use of project resources together with their associated project member users, effectively reducing the possible budget and time constraints associated with the project. Accordingly, this defines the ratio of time taken by distinct activity to the estimated time costs.

- **Easy mapping of roles**

The proposed model presents a simpler means through which the developers can be able to blend the existent agile practices into the traditional V-model. A primary advantage of this integration is the direct mapping of roles without the need for the development of additional roles. This means that the proposed model allows the inclusion of the positive roles from each method, leading to the reduction in the ambiguity of the operational procedures of the team members.

6. Conclusion and Recommendations

Software development and integration is a multifaceted form of computing. The availability of functional and efficient software tool for review of trends and activities can result in business success. The V model is one of the software development processes, however, the proposed software development process is more efficient than the V model. The case studies under review have indicated that the proposed model is better than the V-model, since it can handle reduced Testing Time, debugging, prioritization of requirements, easy mapping of roles, and improved visibility of project resources. Additionally, the validation checks are better implemented in the proposed systems.

In terms of cost, the proposed system is less costly compared to the V-model hence eliminating unnecessary financial challenges. The proposed model ought to be implemented and integrated to any systems. The recommendations below are ideal for the companies that are keen on capitalizing on the minimal financial resources under disposal for software design. Companies should employ resources who have experience with agile methods, this will help in propagating his/her knowledge to adopt the agile development. Also one should change his/her mind set to accept this new era of software development, hence there will be less external or internal pressure to follow traditional V model phases and practices. Not to forget the fact that a management support is the key element for a project success. If employees feel well treated in their place of work, the issue of unwillingness of team to follow a new concept of agile development will be eradicated. Improved teamwork and increased productivity will be favored. Last but not the least, Training is very important. It is considered to be the core objectives that a person should acquire before he/she can master the agile v model concept. It is an investment that will very profitable to the company. There is no loss on a long time basis as the resources will be more completive and fully knowledgeable.

References

[1] Ambler, Scott. (2007) "Agile testing strategies". *Dr. Dobb's Journal*.

[2] Beal, Adriana. (2014) "Traditional vs. Agile: The future is Hybrid". Available at: http://www.bridging-the-gap.com/traditional-vs-agile-the-future-is-hybrid/ [Accessed 28 November 2017].

[3] Bucanac, Christian. (1999) "The V-Model". Available at http://www.bucanac.com/documents/The_V Model.pdf

[4] Firesmith, Don. (2013). "Using V Models for Testing". Available at: http://blog.sei.cmu.edu/post.cfm/using-v-models-testing-315 [Accessed 15 November 2017].

[5] Kostic, Dusko & Josimovic, Ljubisa. (2013) "Basic Management Characteristics of Software Engineering", *Fascicle of Management and Technological Engineering*, vol. **2**, pp. 107-111.

[6] Misra, Subhas Chandra. (2007) "Adopting agile software development practices: Success factors, changes required, and challenges". Available at http://search.proquest.com/docview/304888117?accountid=12085.

[7] Monteleone, Mark. (2014) "A Proposal for an Agile Development Testing V-Model". Business *Analyst Community & Resources | Modern Analyst*. Available at:http://www.modernanalyst.com/Resources/Articles/tabid/115/ID/1967

[8] Pelrine, Joseph. (2011). "On Understanding Software Agility- A Social Complexity Point of View". *E:CO* Issue Vol. **13** Nos. 1-2 2011 pp.26-37

[9] Regulwar Ganesh, Deshmukh Prashant, Tugnayat Rohit, Jawandhiya Pradip and Gulhane Veena. (2010). "Variations in V model for software development". *International Journal of Advanced Research in Computer Science,* **1**(2) Available at http://search.proquest.com/docview/1443699838?accountid=12085

[10] Rouse, Mike (2007). "What is V-Model (Vee-Model)?" Available at: http://searchsoftwarequality.techtarget.com/definition/V-Model [Accessed 15 November 2017].

[11] Turk Dan, France Robert and Rumpe Bernhard. (2005). "Assumptions underlying agile software-development processes*". Journal of Database Manage*ment, **16**(4), 62-87.

[12] Venantius Laulin Madhu (2014). "Agile Testing: Key Points for Unlearning". Available at: https://www.scrumalliance.org/community/articles/2012/january/agile-testing-key-points- for-unlearning [Accessed 28 November 2017].

[13] Verwijs, Christiaan. (2012). "ChristiaanVerwijs.nl | 5 Principles for (Agile) Software Development that improve Agility (and make you a better developer)."Available at: http://www.christiaanverwijs.nl/post/2012/10/04/5-Principles-for-(Agile)-Software-Development-that-improve-Agility-(and-make-you-a-better-developer).aspx [Accessed 15 November 2017].