International Journal of Advances in Engineering Research http://www.ijaer.com/

(IJAER) 2012, Vol. No. 3, Issue No.IV,April **ISSN: 2231-5152**

CAPABILITY DETERMINATION OF SOFTWARE DESIGN USING V- MODEL

Prof. Sheo Kumar

ABSTRACT

Software development is about building useful systems, not generating reams of documents. The V-Model helps the development team apply focus to what documents are useful (and why) and how much content is appropriate for each. The V-model offers a framework that clarifies the relationships between requirements, specifications,andtesting.ThispaperdiscussesthebenefitsoftheV-Modelinavarietyofdiffering development processes and capability determination including waterfall and agile.

KEYWORDS: V-model, Software Testing,Software Engineering, Software architecture, Software Development Life cycle, Static testing, Dynamic testing.

INTRODUCTION

Developing software can be difficult enough without development managers or project managers demanding we produce reams of documentation too. If there's a clear purpose to writing something down then the task becomes more palatable especially useful if we don't feel that we're natural authors or writers. The V Model attempts to put some structure and purpose into the documents that we should consider producing during the lifetime of our development project. Whilst it is not a process itself, it helps establish a standard process for creatingsoftware.

The V-model is a software development model which can be presumed to be the extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. The V-Model demonstrates the relationships between each phase of the development life and its associated phase oftesting.

Today the IT Solutions & Products involve large investments and critical data of the organization concerned. During development and maintenance of such long lived software, requirements are analyzed, designed and code modules are developed, testing is planned and code is tested many times. Thus software development and maintenance services should ensure customer satisfaction. This calls for software developer to ensure the quality of development, implementation, testing and as wellas

maintenance. Since the schedule of software development is running tight, resulting in less effort for testing and maintenance. As testing directly links to quality of product, this demands that solution provider creates strong Testing and Maintenance Base for the technologysolutions.

Software Testing is the most important phase of the Software Development Life Cycle. On most software projects testing activities consume at least 30 percent of the project effort. On safety critical applications, software testing can consume between 50 to 80 percent of project effort. Software testing is essential to ensure software quality. Schedule is always running tight during the software system development, there after reducing efforts of performing software testing management. In such a situation, improving software quality becomes an impossible mission It is our belief that software industry needs new approaches to promote software testing management. The article discussed the model that already existed, further excavates the parallelism between test stages and maintenance test stages and tries to propose a improved V model. This model makes the software testing pass through the each stage of software development cycle. That can discover software mistakes as early aspossible.

TRADITIONAL V-MODEL

The V-model is a software development process which can be presumed to be the extension of the waterfall model. It was the first proposed by Paul Rook in the late 1980s and is still in use today. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical Vshape.

The V-model deploys a well-structured method in which each phase can be implemented by the detailed documentation of the previous phase. Testing activities like test designing start at the beginning of the project well before coding and therefore saves a huge amount of the project time. The purpose of V model is to improve efficiency and effectiveness of software development and reflect the relationship between test activities and development activities as shown inFigure

1. V-model is perhaps the most traditional model followed for management of software tests.

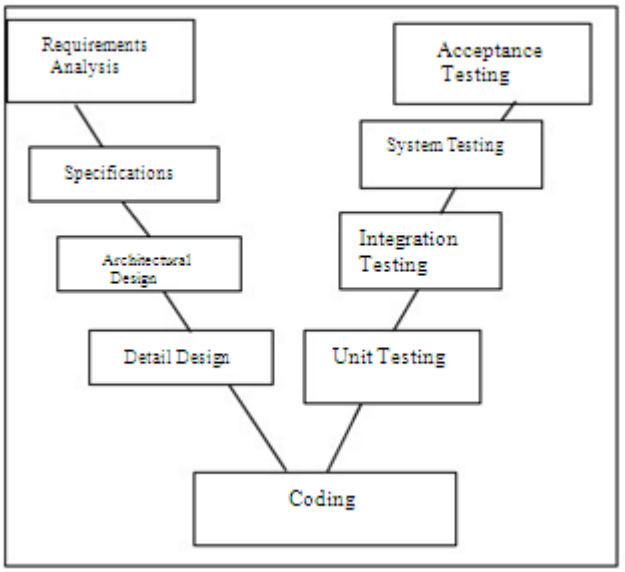

Figure 1. V Model

The basic V Model development process is divided into understanding of user's requirements, performing requirements analysis, designing the initial outline, designing advanced detailed and describing required tests in the basic development process as shown in figure 1 from left to right on each other counterparts. Testing is too important to leave to the end of the project, and the V-Model of testing incorporates testing into the entire software development life cycle. In a diagram as in Figure 1 of the V-Model, the V proceeds down and then up, from left to right depictingthe basic sequence of development and testing activities. The model highlights the existence of different levels of testing and depicts the way each relates to a different development phase. Like any model, the V-Model has detractors and arguably has deficiencies and alternatives but it clearly illustrates that testing can and should start at the very beginning of the project. In the requirements gathering stage the requirements are gathered, verify and validated inorder to justify the project. The business requirements are also used to guide the user acceptance testing. The model illustrates how each subsequent phase should verify and validate work done in the previous phase, and how work done during development is used to guide the individual testing phases. This interconnectedness lets us identify important errors, omissions, and other problems before they can do seriousharm.

NEW MODEL

V model is the most representative model for traditional software testing

management. The purpose of the V model is to improve efficiency and effectiveness of software development and reflect the relationship between test activities, development activities and maintenance activities. Once the system has been made functional and all activities have been performed, if it is not maintained properly, all the development and testing efforts shall go in vain. Thus in this section of the paper we propose a new improved V model called as the Advanced V model that reflects the relationship between the development activities, test activities and maintenance activities in order to achieve a highly efficientand reliable system.

ADVANCED V MODEL

Software testing is described as a continuous improvement process that must be integrated in into an application maintenance methodology. The term software maintenance usually refers to changes that must be made to software after they have been delivered to the customer or user. Software testing and software maintenance are the most important phases of software development life cycle that go hand in hand to obtain reliable software. The Advanced V model of testingincorporates testing and maintenance activities into the entiresoftware development life cycle. In the diagram as in Figure 2 of the Advanced V-Model, it proceeds down and then up, from left to right depicting the basic sequence of development, testing and maintenance activities. The model highlights the existence of different levels of testing with respect to their maintenance activities tests and depicts the way each relates to different development phase activities. The testing commences together with the initial phase of development oftheproject. In the requirements gathering stage the requirements aregathered, analysed, verified and validated in order to justify the project.

The business requirements at the same time also guide to the acceptance testing. Once the acceptance testing is done the error free product needs to be deployed as per the satisfaction of thecustomer.

The Advanced V Model development process is divided into understanding of the user's requirements, performing requirements analysis, specification, designing the initial and detailed outline and laying out the program specifications. Then the required tests are described in the basic development process as shown in figure 2 from left to right on each other counterparts along with the maintenance tests being carried out for each of the test activity as shown in the figure. Once the activities of the development phase starts simultaneously the activities of the testing phase and maintenance phase commence. Ever imagined software being deployed without carrying out these testing activities? No would be the prompt reply. So with the unit testing, the modules programmed are tested and test cases are designed. Then moving on to the next level is integration testing where individual modules are integrated and tested for functionality. But this is incomplete without regression testing as the updated changes are then reflected.System

testing describes the testing of the system as a whole. Along with it we need to do the security testing in order to check the systems compliance to various security threats. In the modern era where technology is moving with the speed of light, the needs to deploy security measures have increased. Thus security threats like unauthorized access, user permission grant needs to be checked at each phase of development activity and testing activity. Then once the user is satisfied after conducting the alpha and beta tests of acceptance test activity the software or the product is deployed at the customers place. Thus a continuous interaction of the development activities, testing activities and maintenance test activities completes the software development life cycle of the product thereby efficiently carrying out the software test managementactivities.

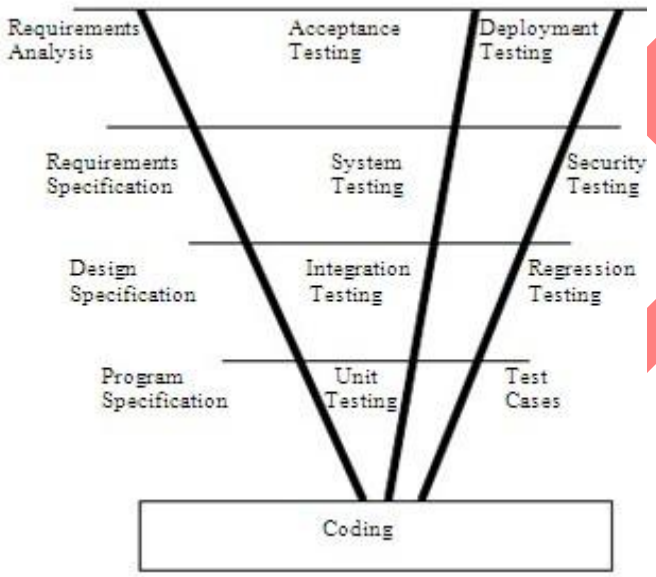

Figure 2. The Advanced VModel

ARCHITECTURE OF THE ADVANCED VMODEL

Architecture gives the structural description of the components and thereby helping us to understand the components in a modified and better way. This section of the paper will describe the architecture of the component for software testmanagement.

COMPONENT STRUCTURE DIAGRAM

In this section we construct a software test management structure of component from a structural point of view. The structure of the components of software testing managementandsoftwaremaintenancetestsarethebasicelements,andtheycomposeof

software testing management structure.

The necessary and beneficial structural components are analyzed from software development, testing and maintenance point of view. That is, we need to identify all the builders and destroyers of the software testing management such as customers and the role of the users are asfollows:

1. **Project Manager:** A project manager is a facilitator. The project manager is the one who is responsible for making decisions in such a way that risk is controlled and uncertainty minimized. Every decision made by him should ideally be directly benefit the project. He must possess a combination of skills including the ability to ask penetrating questions, identify unstated assumptions, and resolve personnel conflicts along with more systematic managementskills.

2. **Software Development Manager:** Leads a team of programmers. Development Manager is responsible for leading the software development team in support of the software development life cycle process, change management, development environments and production releases. He will provide overall supervision and technical guidance to the development team in understanding requirements, preparing high level and low-level designs, coding and building thesoftware.

3. **Software Architect:** An architect acts as a technically savvy business owner. He deals with the interactions of systems, whether between components written in different languages at different times and at different locations, or between components of the same software system that use the same coding language. Architects deal with the interactions of systems, whether between components written in different languages at different times and at different locations, or between components of the same software system that use the same codinglanguage.

4. **Software Developer:** A software developer is a person concerned with facets of the software development process wider than design and coding, a somewhat broader scope of computer programming or a specialty of project managing including some aspects of software productmanagement.

5. **Test Manager:** Test managers really serve two different customers, their testers and corporate management. For the testers, he helps develop product test strategies, and provides test expertise to the testing group. For management, he gathers product information so that corporate management can decide when the product is ready for implementation.

6. **Test Leader:** Technical leader acts as in interface between the test manager and the testing team. He is responsible for the completion of the testing as per the designed time frame.

7. **Test Designer:** Test designer is responsible for developing test strategies and test plans. He provides an assessment on the overall status of the testing program. He stays well informed and connected with the industry and the current trends in the technologies available.

8. **Software Tester:** Software tester is responsible for carrying out software testing using various strategies of testing. He builds up the test cases and test plans for theproject.

9. **Quality Manager:** Quality Manager works towards customizing software development processes. He is responsible for creating and implementing a quality management program plan for the entire organization and works towards process improvement.

10. **Quality Assurance Engineer:** Software quality assurance engineer deals with the location of the defect and mechanisms to preventdefects.

11. **Quality Control Engineer:** Software quality control engineer looks after the set of activities designed to evaluate the quality of developedsoftware.

12. **Quality Guarantee Engineer:** Software quality guarantee engineer is responsible for maintaining software quality. He is responsible for tackling and solving all softwareproblems.

TARGET

We need to produce software that meets thecustomers' expectations. We can summarize those expectationsas:

- Functional – it allows the customers to get their day jobsdone
- Operable on a day-to-day basis – the behind the scenes and/or automated tasks can be doneeffectively
- Maintainable-problemscanbefixed,andenhancementscanbemade
- Costeffective–alloftheabovecanbedoneatanaffordableand

justifiable price

To meet those demands we are going to need to test the code. To test it we are going to need to know what it's meant to do. Knowing what it's meant to do implies that the requirements and design are written down somewhere, but we don't need to get into any arguments over documentation; as we'll see later in this paper, comments in code can adequately substitute for formal documents if the comments are well written (this paper is agnostic on the subject of waterfall versus agile,etc.).

We need a development framework that ensures we collect testable requirements and design.

THE SOLUTION MEET

THE V

We can simplify software development as a series of tasks thus:

a) Customers tell us what theywant,

b) We figure-out how we can deliver that, c) We buildit,

d) We make sure we built what we intended to,and

e) The customers check that we've built something that does what they need.

This linear progression can be bent and folded in many ways in order to describe lots of different processes; so, for instance, in iterative projects we may run through the sequence many times, adding extra bits of functionality each time; if we use test driven design we might do some of the steps out of sequence, but at the end of any phase of development we will have completed elements of all five steps a through e. So, in essence, we have to have the steps that we havelisted.

The V-model simply takes those five steps and places them onto a V-shape thus:

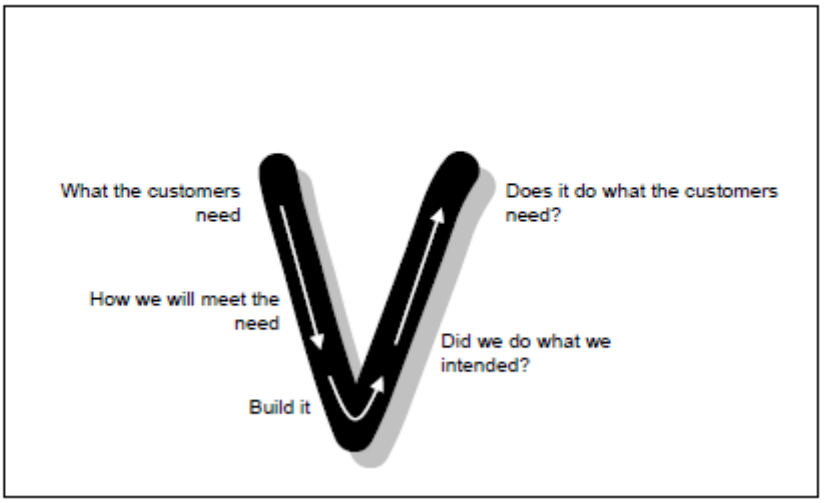

Figure 3. The Basic Flow Through The V

There's a relationship between the elements on the left-hand side. They represent the decomposition of the problem from business-speak into technical speak and ultimately into technical activity (from top to bottom). This is sometimes known as elucidation or verification.

We can see that the right-hand side of the V contains the tests of the counterparts on the left-hand side, e.g. "Does it do what the customers needed?" is the test of "What the customers need". This is sometimes known as testing or validation. So the left and right sides of the V capture verification and validation.

Let's expand those activities:

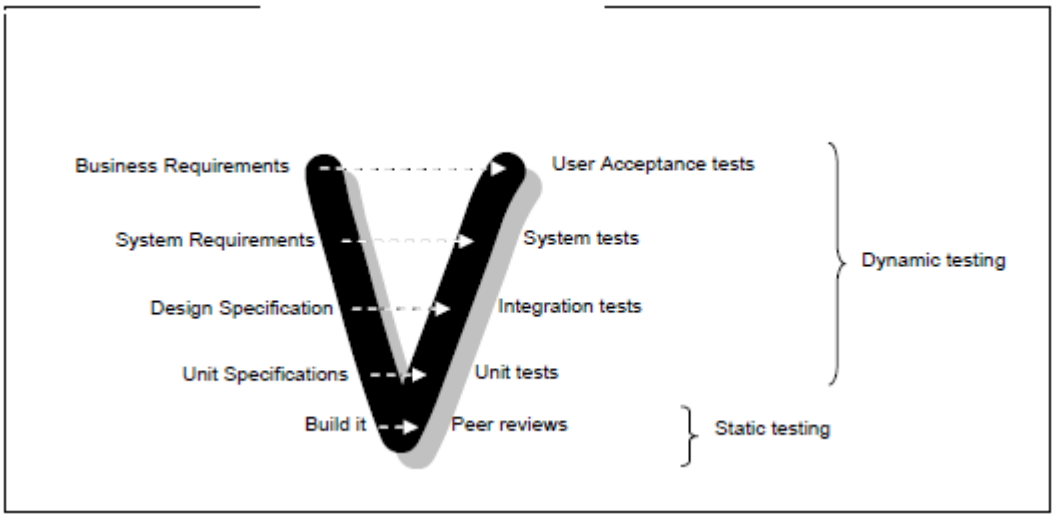

Figure 4. Traceability to testing

CONCLUSION

The V-Model is a framework, not a process. It can be used with a wide variety of the popular development processes, from water fall to agile. It does not specify what documents are required, it merely provides guidance on what information should be collected, why that information should be collected, and how that information relates to other information (whether it be relating to higher or lower degrees of elucidation/verification, or whether it relates totesting/validation).

Adopting the V-Model framework as part of our development process:

- facilitates greater control due to standardization of products in the process
- delivers an increase in quality due to the ease with which templates and examples can be produced and due tofamiliarity with the process's products
- easier cost estimation due to the repeatability of theprocess
-

The major contribution of this research is in applying software architecture based on component structure diagram for efficient software testing management. We propose an Advanced V model describing that for efficient software testing management along withthedevelopmentandtestingprocess,themaintenanceprocessisalsoequally

important. Thus we have integrated these processes for efficient software testing management.

We have achieved what should be done, why should be done and how it should be done in software testing management at all the phases of the software development. Maintaining the software before and after testing helps improving the quality of the software to a large extent. Our approach provided important guidelines to the developing software industry where technology keeps on changingeveryday.

Adopt the V-Model. V is for victory!

REFERENCES

[1] G.Rothermel et al "Test Case Prioritization: An Empirical Study" Proceedings of the International Conference on Software Maintenance, Oxford,UK, September, 1999

[2] S. Elbaum, A. Malishevsky,and G. Rothermel.Prioritizing test cases for regression testing.Proc. Int'1 Symp. Softw. Testing and Analysis, Pages 102-112, Aug.2000.

[3] D.Leon and A. Podgurski. A comparison of coveragebased and distribution- based techniques for filtering and prioritizing test cases. In Int'1. Symp. Softw. Rel. Eng., Pages 442-453,Nov.2003.

[4] J.kim and A. Porter. A history-based test prioritization technique for regression testing in resource constrained environments. In Int'1. Conf. Softw. Eng.,Pages 119-129, May 2002.

[5] D.Jeffrey and N. Gupta. Test case prioritization using relevant slices. In Int'1. Comp. Softw. Appl. Conf., pages 411-420,Sept.2006.

[6] RenéeC.Bryce&AtifM.MemonDoSTA'07,September4,2007,Dubrovnik, Croatia.ACM

[7] Z.Li, M.Harman,andR.M.Hierons. Search algorithms for regression test case prioritization.IEEE Trans. Softw. Eng., 33(4):225-237,Apr.2007.

[8] Pavan Kumar Chittimalli& Mary Jean Harrold ISEC'08, February 19-22, 2008, Hyderabad,India.

[9] Lijun Mei, Zhenyu Zhan et al WWW'09,April 20-24,2009,Madrid,Spain.ACM 978-1-60558-487-4/09/04.

[10] KristenR.Walcottetal,"TimeAwareTestSuitePrioritization",ACM, ISSTA'06,, Portland, Maine,USA, July 17-20,2006

[11] B. Qu, C. Nie, B. Xu and X. Zhang. Test Case Prioritization for Black Box Testing. In Proceedings of the International Computer Software and Applications Conference, pages 465-474, July,2007.

[12] Qurat-ul-annFarooq, Muhammad Zohaib Z. Iqbal, Zafar I Malik," An Approach for Selective State Machine based Regression Testing", ACM 978-1-59593- 850-3/07/0007,AMOST'07,London,UK.July 9-12,2007

[13]Wang L., J.Yuan,X.Yu, J. Hu, X. Li,and G. Zheng. Generating Test Cases from UML Activity Diagram based on Gray-Box Method. In 11th Asia-Pacific Software Engineering Conference (APSEC 2004