International Journal of Information Technology and Business Management
29th June 2012. Vol.2 No. 1


ISSN 2304-0777 www.jitbm.com

WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC

S.Balaji
Computer Science Dept., Gulf College
Muscat, Sultanate of Oman.

Dr.M.Sundararajan Murugaiyan
Computer Science Dept., Government Arts College
Chennai, TN, India.

Email: sundramoorthybalaji@gmail.com

Abstract**:** Organizations that are developing software solution are faced with the difficult choice of picking the right software development life cycle (SDLC). The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The V-model represents a software development process which may be considered an 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 Agile Modeling is a practice-based methodology for modelling and documentation of software-based systems. It is intended to be a collection of values, principles, and practices for modelling software that can be applied on a software development project in a more flexible manner than traditional Modelling methods. This comparative summarizes the steps an organization would have to go through in order to make the best possible choice.

Keywords: SDLC, Waterfall, V-Model, Agile

1. INTRODUCTION

A Software Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below. A number of software development life cycle (SDLC) models have been created: waterfall, spiral, V-Model, rapid prototyping, incremental, and synchronize and stabilize. Waterfall model is the Sequential development model. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. Agile software development [4] is a group of software development methodologies based on iterative and incremental development [6], where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams

2. Q&A BEFORE DECIDING WHICH MODEL TO BE USED

Before deciding the model to be used, we should get answer for some questions.

1. How stable are the requirements?
2. Who are the end users for the system?
3. What is the size of the project?
4. Where are the Project teams located?

2.1 Case Study

Our Scenario is to discuss for the requirement given by the client which development Life cycle method to be used. Let us have a comparative study which model will be effective in the below models and the Pros & Cons of choosing the model.

1. Waterfall Model
2. V-Model
3. Agile Model

International Journal of Information Technology and Business Management
29th June 2012. Vol.2 No. 1

ISSN 2304-0777 www.jitbm.com

The client Requirement is to develop a web based application, online booking of ticket for Train, bus and flight in single site. The Challenging in developing this site. User can able to book for next six month. In existing system web site will not allow to book for next six months. In Proposed System user can able to login to the site with valid authentication. If they want to book train ticket after three month they can able to fill all the details in the site, payment and it will save in database. Ticket will be automatically booked when booking open for that particular date and month. So this requirement will help lot of User to access this site.

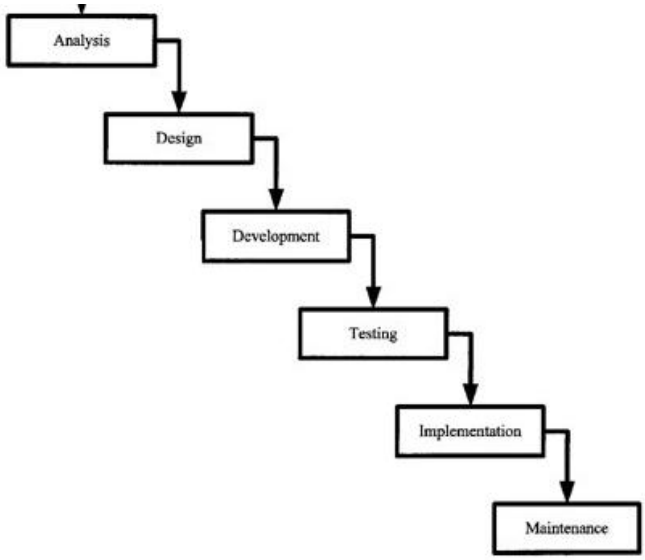
In Our comparative study, we are going to discuss about which model to choose for best SDLC to deliver the quality product to the client.

3. COMPARATIVE STUDY OF WATERFALL VS V-MODEL VS AGILE PROS AND CONS

3.1 Waterfall Model

- Waterfall model [16] is the Sequential development model.
- Requirement should be clear before going to next phase of design.
- Testing is carried out once the code has been fully developed. Each work-product or activity is completed before moving on to next
- Each phase of development proceeds in order without any overlapping.
- Each phase schedule for the tasks to be completed within a specified time period
- The documentation and testing happens at the end of each phase, which helps in maintaining the quality of the project.
- In the waterfall model each step is frozen before the next step. That is the requirements are frozen before the design starts, and once the design is frozen the coding starts etc. But what will the testing team do till then so is very time consuming and high costing
- In waterfall model the defect were found very late in the development life cycle as test team was not involved from the beginning of the project.
- Tester role will be involved in testing phase only

Figure 1: Waterfall Model Life Cycle



Requirement given by the client should be clear before we start the next phase of development life cycle because in waterfall model the requirement phase should be freeze before we start the design phase. Further changes in requirement will not be considered.

3.1.1 Pros

- Requirement is clear before development starts.
- Each phase is completed in specified period of time after that it moves to next phase.
- As its linear model, it's easy to implement.
- The amount of resources required to implement this model are minimal.
- Each phase proper documentation is followed for the quality of the development.

3.1.2 Cons

- The problems with one phase are never solved completely during that phase and in fact many problems regarding a particular phase arise after the phase is signed off, this result in badly structured system.
- If client want the requirement to be changed , it will not implemented in the current development process

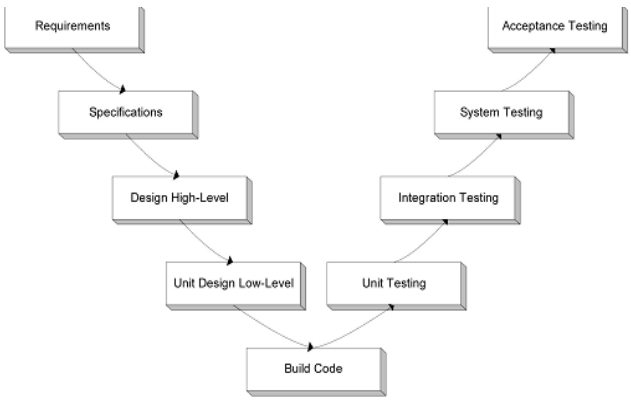
ISSN 2304-0777 www.jitbm.com

In spite of the cons, the many pros of this model ensure that it remains one of the most popular models used in the field of software development.

3.2 V-Model

- The V model (Validation & Verification model)[11] is a modified version of the Waterfall method
- As opposed to the Waterfall method, this one was not designed in a linear axis; instead the stages turn back upwards after the coding phase is done.
- This developmental process is balanced and relies on the verification from the previous steps before proceeding forward.
- The product from every phase needs to be checked and approved before moving forward.
- In v model developer and tester works parallel.
- In V model, based on the requirements the System test cases are prepared, and based on the HLD (High level document)the Integration Test cases are prepared, and based on the LLD(Low-level document)the Integration Test cases are prepared. And then the coding is done. Once coding is completed, unit, integration and system testing happens in the sequence.
- In V-model, gives relationship between each development stages and Testing stages.

Figure 2: V- Model Life Cycle



Note: The left tail of "V" represents 'specifications phase', the right tail of "V" represents 'testing phase', the bottom of the "V" where tails meet represents 'development phase'.

3.2.1 Pros

- Same as Waterfall model
- V-Model, the advantage is that Tester role will be involved in the requirement phase itself.
- Requirement Changes is possible in any phase.

3.2.2 Cons

- The biggest disadvantage of V-model is that it's very rigid and the least flexible.
- If any changes happen mid way, not only the requirements documents but also the test documentation needs to be updated.
- It is not proposed for short term projects as it requires reviews at each stage.

In our case study requirement , if client need to change any requirement its possible to update but documentation prepared from requirement phase like functional Specification , high level design, low level design ,unit testing, system testing ,integration testing to be updated.

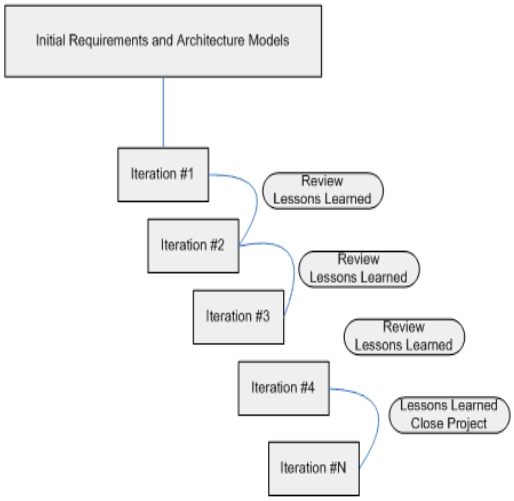
Mostly V-Model is used in Larger Organization as it requires more number of resources.

3.3 Agile Model

- The term agile stands for 'moving quickly'
- Agile methodology [12] has an adaptive team which is able to respond to the changing requirements.
- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Working software is delivered frequently (weeks rather than months)
- The most important of the principles is customer satisfaction by giving rapid and continuous delivery of small and useful software

ISSN 2304-0777 www.jitbm.com

Figure 3: Agile Model Life Cycle



3.3.1 Pros

- The most important of the advantages of agile model is the ability to respond to the changing requirements of the project
- There is no guesswork between the development team and the customer, as there is face to face communication and continuous inputs from the client

3.3.2 Cons

- If the projects are smaller projects, then using the agile model is certainly profitable, but if it is a large project, then it becomes difficult to judge the efforts and the time required for the project in the software development life cycle.
- Only senior developers are in a better position to take the decisions necessary for the agile type of development, which leaves hardly any place for newbie programmers, until it is combined with the senior resources.

4. CONCLUSION

As we discussed on Waterfall [16] V-Model [13] & Agile model [12] Pros and Cons, it depends upon the organization which model to choose.

- If requirement changes frequently and smaller projects, deliver product in short period time with skilled resources then we can choose "Agile model ".
- If requirement is clear, larger project then we choose "Waterfall model"
- If requirement changes , larger project , proper validation to take place in each phase , tester to be involved in early stages of development, then we can choose "V-Model".

REFERENCES

1. Ambler, S. (2002). "Agile Modeling: Effective Practices for XP and RUP.".
2. Beck, K. &. (2001). "Manifesto for Agile Software Development". Agile Alliance.
3. Beck, K. (2003). "Test-Driven Development by Example".
4. Cockburn, A. (2006). "Agile Software Development (second edition)".
5. D.North. (2006). "Introducing Behaviour Driven Development".
6. David Cohen, M. L. (2003). "Agile Software Development",Data & Analysis Center for Software.
7. Forsberg, K. a. (1991). "The Relationship of Systems Engineering to the Project Cycle," First Annual Symposium of the National Council On Systems Engineering (NCOSE).
8. K, P. (2010). "Doctoral research in Sweden Implementing Lean and Agile Software Development in Industry".
9. Kent Beck, M. F. (2000). "Planning Extreme Programming". .
10. Larman, C. (2004). "Agile and Iterative Development: A Manager's Guide. Addison-Wesley".
11. Limits of the VModel (1997).
12. Marti, R. C. (1999). "Agile Software Development – Principles, Patterns and Practices".
13. Overview of the Activity Model of the V-Model. . (1997).
14. Palmer, S. (2002). " A Practical Guide to Feature-Driven Development. Prentice Hall".
15. Schwaber, K. (2002). M. B. "Agile Software Development with Scrum".
16. Weisert, C. (2003). Waterfall methodology: there's no such thing.

ISSN 2304-0777 www.jitbm.com