

New Idea In Waterfall Model For Real Time Software Development

Unnati A. Patel^a, Niky K. Jain^b

a Assistant Professor, M.Sc (IT) Department, ISTAR, Vallabh Vidya Nagar, Gujarat

b Assistant Professor, M.Sc (IT) Department, ISTAR, Vallabh Vidya Nagar

Abstract:

This article presents modification in Waterfall Model for Software Development. Waterfall model is base software development model which is most used to develop a software in software development companies. In this paper, the brief definition of waterfall model is given. Many problems have been faced by software companies by using this model. Therefore, the main objective of this research is to represent some modifications to the model, which correct most of the problems, connected with the waterfall model. Commonly accepted problems are for example to survive with change and that defects all too often are detected too late in the software development process. Another problem is related to prototype i.e. user can only view the proposed system after testing. If users are not satisfied with the proposed system then it is too late to redesign it and the time and cost for implementation and testing is totally wasted on selected design.

Keywords- software development process, software development model, software development, SDLC (Software Development Life Cycle), SE (Software Engineering), V&V (Verification & Validation), QA (Quality Assurance)

1. INTRODUCTION

1.1 Introduction to SE

SE is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software. SE processes are composed of many activities, especially the following:

1. Requirements Analysis & Specification
2. Software architecture
3. Implementation
4. Testing
5. Training and Support
6. Maintenance

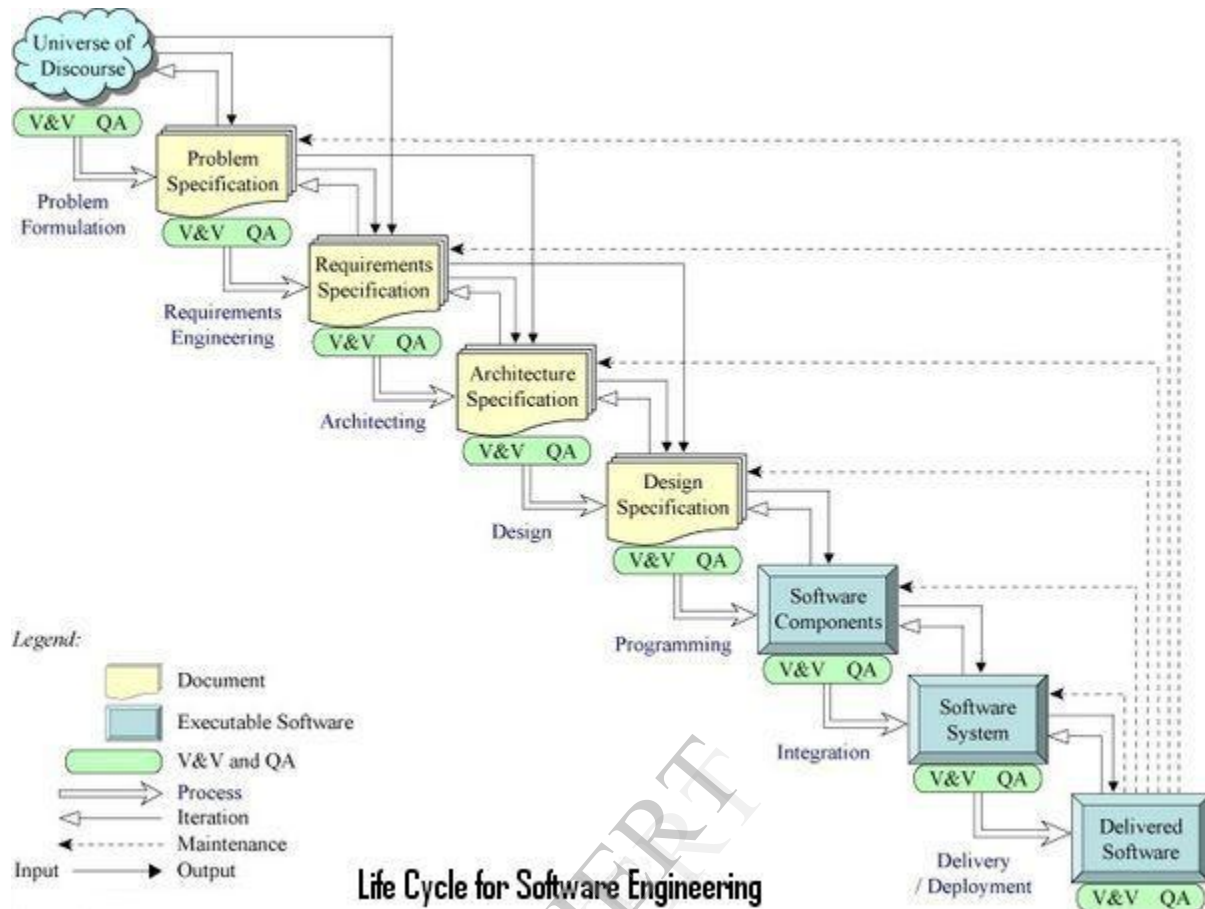


Figure 1: Life Cycle for Software Engineering

A software development process, also known as a SDLC shown by figure 1, is a structure imposed on the development of a software product. It is often considered as a subset of system development life cycle. There are several models for such processes, each describing approaches to a variety of activities that take place during the process. E.g. Waterfall Model.

Problem solving in software consists of these activities:

- Understanding the problem
- Deciding a plan for a solution
- Coding the planned solution
- Testing the actual program

For large systems, each activity can be extremely complex and methodologies and procedures are needed to perform it efficiently and correctly. Furthermore, each of the basic activities itself may be so large that it cannot be handled in single step and must be broken into smaller steps.

1.2 Introduction to Waterfall Model

The first publication on the waterfall model is credited to Walter Royce's article in 1970. New models were developed and older ones were criticized and modified. The standard waterfall model for systems development is an approach and based on the assumption that big decisions have to be made before the coding begins, since cost of development rises exponentially by phases.

Waterfall approach was first Process Model to be introduced and followed widely in SE to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate process phases.

The phases in Waterfall model are:

1. Requirement Specifications phase (Identification and Analysis of the system requirements)
2. Software Design (High Level Design & Detail Level Design)
3. Implementation (Coding, Debugging and Unit Testing)
4. Testing (Integration Testing and System Testing)
5. Maintenance. (System is deployed, ready for operation and Maintained if needed)

1. Requirement Specifications phase

Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements like; Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

2. Software Design

In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

3. Implementation

On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the SDLC.

4. Testing

After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.

5. Maintenance

After successful testing the product is delivered / deployed to the customer for their use. After deployment maintenance phase is started.

All these phases are cascaded to each other so that second phase is started as and when defined set of goals are achieved for first phase and it is signed off, so the name "Waterfall Model".

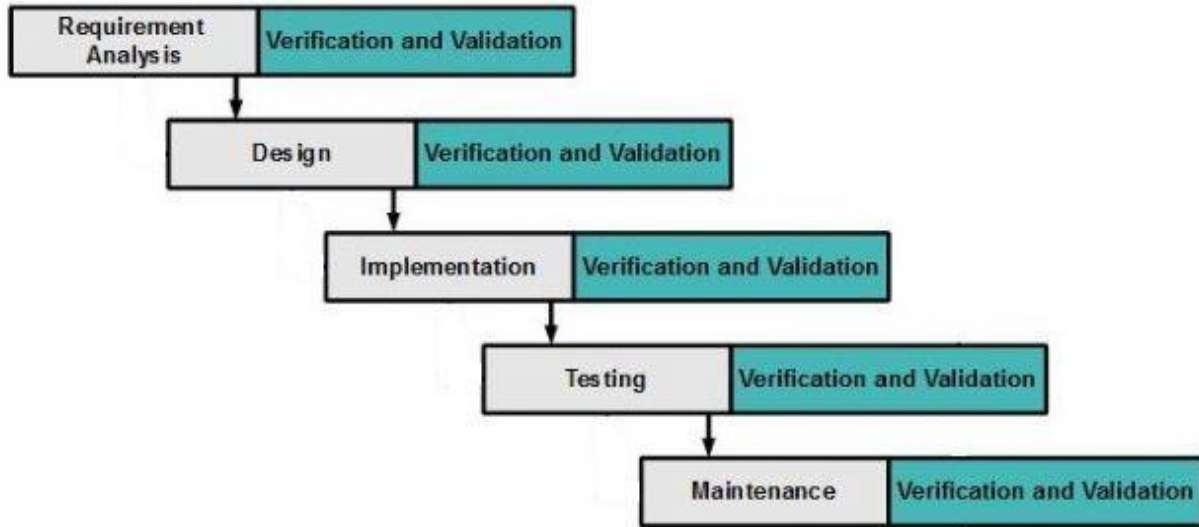


Figure 2 : Original Waterfall Model

2. Criticism Of The Waterfall Model

Since the waterfall model has been around for over 30 years, it has been tested, to the large extent, and number of problems has surfaced.

Critics complain that the waterfall model is

1. Unstable - since client requirements have to be fixed before the development process begins any requirement changes destabilizes the whole development process.
2. Impossible to view the model – user can only view the system after implementation.
3. Too expensive - since requirement inconsistencies and missing components are discovered during design and coding phases, increasing the development needs and therefore costs.

There is also a widely voiced opinion that the idea of Requirement Fixing before the design and coding starts is too idealistic and therefore impractical.

3. Proposed Waterfall Model

The key to successful use of these new features is rigorous validation of requirements, and verification (including testing) of each version of the software against those requirements after each phase of the model. Each phase starts with a design goal and ends with the client (who may be internal) reviewing the progress thus far.

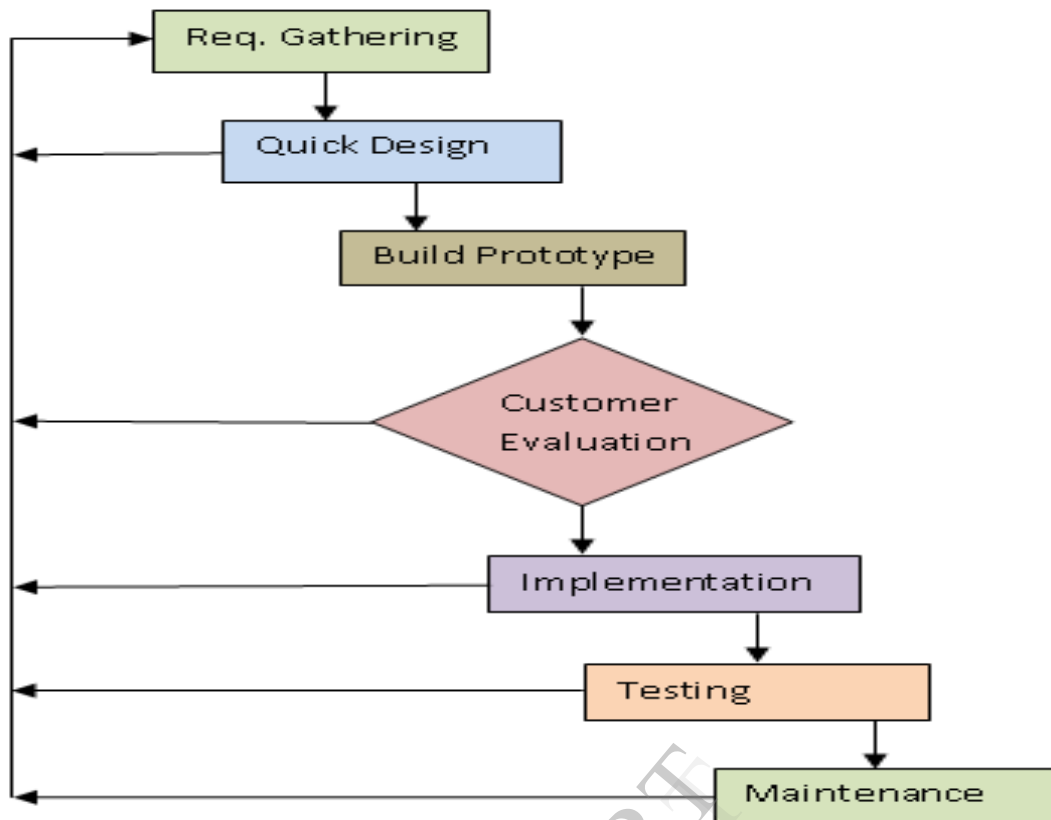


Figure 3: Proposed Waterfall Model

This modified waterfall model has the following strengths:

1. In real industry situation, client requirements may vary at any stages of development. Our proposed modification can allow changing the client requirement at any stage. After each stage there is a review by user, if there is any change in requirement then user can start from the requirement phase for requirement change cycle.
2. User can view the system model before development of the actual system and just after the design phase, so that if there is any change in the model then it can be handled in early phase of development.

3. User can review the ongoing system after each stage, so that if there is any change then it can be incorporated at early stage as soon as it is observed so it saves the cost and time both.

4. Conclusion

After analysis of waterfall model, it has been found that the original water fall model is used by various big companies for their internal projects. Since in the big companies, most of the time clients may change requirements at later phase after partial development of the system, this is the problematic situation for the development team. Proposed water fall model overcome the drawback of original waterfall model. It is of a certain interest to industry, since this model aims to solve most of the critical problems that there in the standard waterfall system.

drawback of original waterfall model. It is of a certain interest to industry, since this model aims

5. References:

1. A. M. Davis, H. Bersoff, E. R. Comer, "A Strategy for Comparing Alternative Software Development Life Cycle Models", Journal IEEE Transactions on Software Engineering ,Vol. 14, Issue 10, 1988
2. Chan, D.K.C. Leung, K.R.P.H, "Software development as a workflow process", 2-5 Page(s): 282 – 291, Dec. 1997.
3. Royce, W.: Managing the development of large software systems: Concepts and techniques. In: Proc. IEEE WESCOM. IEEE Computer Society Press, Los Alamitos (1970)
4. Pfleeger, S.L., Atlee, J.M.: Software engineering: theory and practice, 3rd edn. Prentice Hall, Upper Saddle River (2006)
5. Jones, C.: Patterns of Software Systems: Failure and Success. International Thomson Computer Press (1995)
6. Ian Sommerville, "Software Engineering", Addison Wesley, 7th edition, 2004.
7. Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.
8. Rlewallen, "Software Development Life Cycle Models", 2005 ,<http://codebeter.com>.
9. CTG. MFA – 003, "A Survey of System Development Process Models", Models for Action Project: Developing Practical Approaches to Electronic Records Management and Preservation, Center for Technology in Government University at Albany / Suny,1998 .