Volume 3, Issue 4, October 2016 ISSN: 2348-8565 (Online)

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Software Development Life Cycle

Dr. Sudeep Kishore Sharma
Professor
St. Aloysius Institute of Technology
Jabalpur (M.P.), [INDIA]
Email: sharma.sudeepcs@rediffmail.com

Amaresh Singh
Assistant Professor
St. Aloysius Institute of Technology
Jabalpur (M.P.), [INDIA]
Email: prempsgtech12@gmail.com

ABSTRACT

Software Development Life Cycle is a well defined and systematic approach, practiced for the development of are liable high quality software system. While the software should be inclusive, it should not be unduly cumbersome. Careful attention is required to develop software that is both functional and efficient. Once the software is in use, it must be maintained. In this chapter, we will discuss several processes which can serve as guidelines for software development. The Waterfall model, Incremental Model, Spiral Model and V-Shaped Model are traditional models which follow a set of prescribed steps. The Contemporary models widely used in industries are Rapid Application Development Model, Agile Development Model and Extreme Programming Model.

I. INTRODUCTION

The process of building computer software and information systems has been always dictated by different development methodologies. A software development methodology refers to the framework that is used to plan, manage, and control the process of developing an information system.[1] Software Engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches Figure 1.

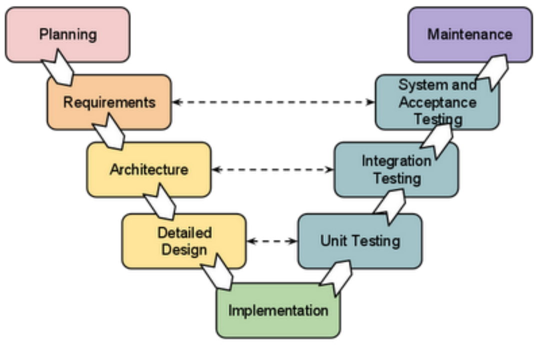

Figure 1.Implementation Cycle

SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality software. [2] The SDLC aims to produce a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates. SDLC is the acronym of Software Development Life Cycle. It is also called as Software development process. The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process [3].ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software [4].

II. SOFTWARE LIFE CYCLE

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process [5].

The following Figure 2 is a graphical representation of the various stages of a typical SDLC.

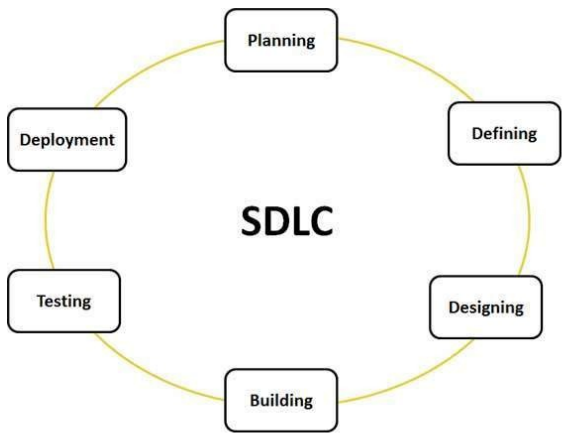

Figure 2 : A typical Software Development life cycle

Phases of SDLC

The various activities carried out for software development can be divided into manageable sections called as phases. These phases may be carried out in different way as per the need. Generally, there are five common phases in SDLC [6]:

(a) Requirement Gathering & Analysis

(b) Designing

(c) Coding

(d) Testing

(e) Maintenance and Support

The requirement gathering and analysis phase helps to understand the problem. This is followed by deciding a plan to solve the problem in the designing phase.[7] The planed solution is then implemented during coding. The Solution program is tested against various test cases. Deployment and maintenance follows this stage. There are various software development approaches designed to develop the software products. These are known as Software Life Cycle Models [8].

SDLC Models

1These models describe the various phases and the order of their execution to be followed to develop the software product. Each phase produces deliverable that are fed as input to the next phase in the life cycle. [9]The product generated by the last phase serves as the final product called as software product. Different models proposed so far are:

A. Traditional models

1. Waterfall Model
2. Incremental Model
3. Spiral Model
4. V-shaped Model
5. RAD Model

B. Contemporary Models

1. Agile Model
2. Extreme Programming Model

III. REQUIREMENTS-GATHERING

During the requirements-gathering phase, the needs of the company are outlined. Managers and users (and in some cases, clients) make their “wish-lists” about what they would like the software to do [10-12]. Analysts ask questions about the intended use of the software, what type of data will be processed, how the software should handle the data, and how the data can be accessed once in the system. Following the requirements phase, the software development team should have a detailed list of functions that the system will perform. Emphasis is on the system's goals, rather than the way in which the system will achieve those goals Figure 3.

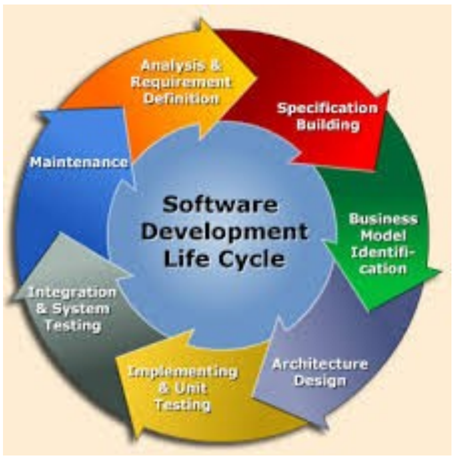
Analysis & Requirement Definition
Specification Building
Business Model Identification
Architecture Design
Implementing & Unit Testing
Integration & System Testing
Maintenance
Software Development Life Cycle

Figure 3: Goal Achievement Cycle

Basic Principle:

- Project is divided into sequential phases, with some overlap and splash back acceptable between phases.
- Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time [13].
- Tight control is maintained over the life of the project via extensive written documentation, formal reviews, and approval/signoff by the user
- Information Technology management to occurring at the end of most phases before beginning the next phase.

IV. Implementation

In the implementation phase, the results of the design phase are translated into program code. Software that does not meet the needs of the company is wasteful. During this phase the programmers should make it their central goal to fulfil the requirements of the company and to meet the design outlined in the design phase. [14] The classes and class interactions developed in the design phase are very explicit. They translate directly into the code generated in the implementation phase. Later in this course, design tools will be introduced that actually automate code generation from the output of the design phase. Spiral model: This model was not the first model [15] to discuss iterative development, but it was the first model to explain why the iteration matters. As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with the client (who may be internal) reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project. The process begins at the centre position. From there it moves clock wise in traversals. Each traversal of the spiral usually results in a deliverable [16]. It is not clearly defined what this deliverable is. This changes from traversal to traversal.

V- shaped lifecycle model

Like the Waterfall model, the V-Shaped life cycle model provides a sequential path of discrete phases that must each be completed before development proceeds to the next phase. The phase test plans are developed during each level of development, and once coding is completed, the tests are conducted. Figure 4is a simplified illustration of the V-Shaped life cycle model [17].

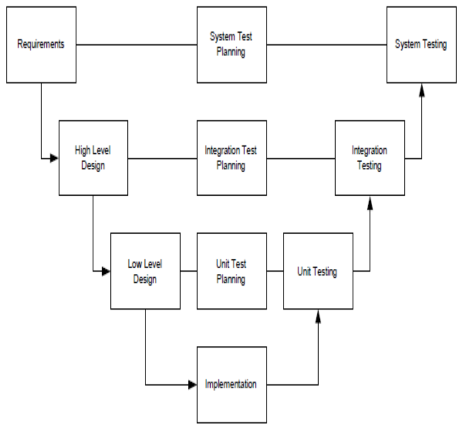

Figure 4 V-Shaped model

Spiral Method (SDM)

It is combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations [18].

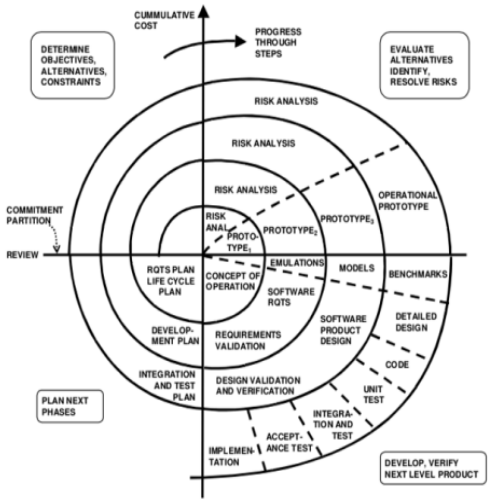

Figure 5 Spiral Model

The usage

It is used in shrink-wrap large applications and systems which built-in small phases or segments.

V. TRADITIONAL MODELS

Traditional software development models are those that are characterized by linear nature, that is, the various phases of SDLC are carried out sequentially. Building the software product initiates with the visualization of the final software, and continues working through to build the final visualized software product [19]. A few traditional models are discussed below.

A. Waterfall Model

Waterfall model was proposed by Royce in 1970. It is known as the classical and the basic model of Software Engineering. It is a linear sequential SDLC model because the various phases are carried out in a sequence. In this model, development is seen as flowing downwards through the following phases [20]:

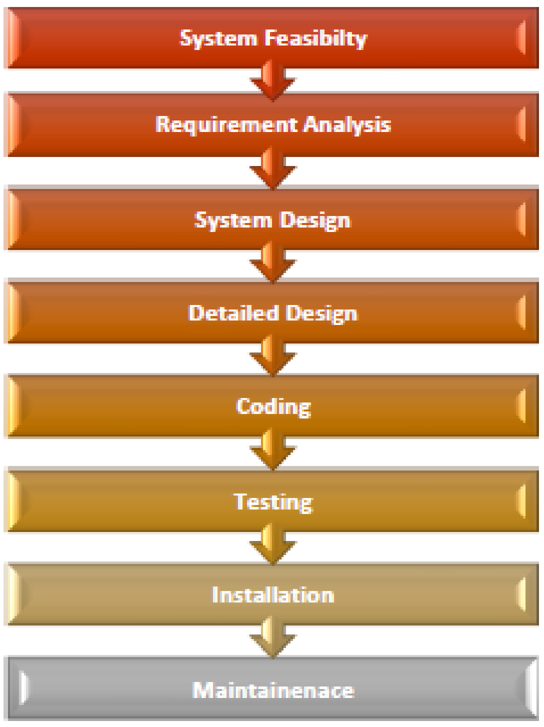

Figure 6: Waterfall Model

VI. CONCLUSION

There are more than tons of SDLC models today. Here only a study those models is given. This paper focused on basic five models; their advantages, disadvantages, so that one can select the best suited model as per his requirements. The Waterfall life cycle model is one of the simplest and easiest to use; it consists of five discrete phases that are executed sequentially. The V-Shaped lifecycle model adds an emphasis on testing to the Waterfall model. The Incremental lifecycle model applies a series of iterations to the Waterfall model. The Spiral life cycle model builds upon the Waterfall and Incremental models and focuses on risk analysis.

REFERENCES:

[1] www.ccs.neu.edu/home/matthias/670s05/Lectures/2.html

[2] Software Engineering Models Consequences And Alternatives Nitin Mishra, Shantanu Chowdhary, Arunendra Singh, Anil Sharma

[3] Study & Comparison of Software Development Lifecycle Models- Gourav Khurana, Sachin Gupta

[4] Gray Pilgrim, "Waterfall Model Vs Agile", Website http://www.buzzle.com/articles/waterfall-model- vs-agile.html, Jan, 201

[5] IanSommerville, Software Engineering, AddisonWesley,9th ed., 2010.

[6] http://istqbexamcertification.com/what-is iterative- model advantages-disadvantages-and-when-to- use-it/

[7] http://www.tutorialspoint.com/sdlc/sdlc_iterative_mod el.html

[8] Raymond Lewallen, "Software Development Life Cycle", 2005

[9] weblog.erenkrantz.com/~jerenk/phase-ii/Boe88.pdf

[10] Mishra A., Dubey D.," A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios", IJARSMS, Volume 1,Page 64-69, October 2013.

[11] Comparative Analysis of Different types of Models in Software Development Life Cycle-Ms. Shikha Maheshwari, Prof. Dinesh Ch. Jain

[12] A Comparison between Five Models of Software Engineering Nabil Mohammed Ali Munassar and A. Govardhan Maheshwari S., Jain Dinesh Ch., "A Comparative Analysis of Different types of Models in Software Development Life Cycle", IJARSMS, Volume 2, Issue 5, May 2012.

[13] Kute S., Thorat S., "A Review on Various Software Development Life Cycle(SDLC) Models", IJRCCT, Volume 3, Issue 7, July – 2014

[14] Abrahamsson P., Warsta J., Siponen M. and Ronkainen J., "New Directions on Agile Methods: A Comparative Analysis", Proceedings of the International Conference on Software Engineering, May 3-5, 2003, Portland, Oregon, USA.

[15] Singh G., Tamanna, "An Agile Methodology Based Model for Software development", IJARCSSE, Volume 4, Issue 6, June 2014

[16] Dora S., Dubey P., "Software Development Life Cycle (SDLC) Analytical comparison and survey on Traditional and agile methodology", NMRJRST, VOLUME NO.2, ISSUE NO.8

[17] http://testingart.com/software-development-life-cycle-models/

[18] http://leansoftwareengineering.com/2008/05/05/boehms-spiral-revisited/

[19] Wallin C., Land R., "Software Development Lifecycle Models The Basic Types"

[20] Ragunath P., Velmourougan S., Davachelvan P., Kayalvizhi S., Ravi Mohan R., "Evolving A New Model

(SDLC Model-2010) For Software Development Life Cycle (SDLC)", IJCSNS, VOL.10 No.1, January 2010

* * * * *