

# A Detailed Analytical Study of Software Process Life

Arvind Kumar Upadhyay

*Professor, Department of Computer Science  
Amity University Madhya Pradesh, Maharajpura Dang, Gwalior (MP) 474005  
aupadhyay@gwa.amity.edu*

**Abstract**—To develop any software, a development strategy is followed. The strategy encompasses the software development life cycle model strategy. In this context, the research paper focuses on applicability of existing development models and suitably recommends a model for current situation. Success of any software depends on a robust development model and it also supports the maintenance activities required for existence of software.

**Keywords**—Software process model, SDLC, Analysis, Design.

## I. INTRODUCTION

The life cycle for software consists of a well planned analysis, design, implementation and maintenance activities. This ensures that the requirement stated by the users is well addressed and taken into account [1]. Also this helps in optimal resource management activities. The methods adopted are useful in determining the software quality satisfying the customer needs [2]. A large software is evolutionary in nature and a combination of development strategies are put together to ensure successful deployment of software and its maintenance. Also the selection of process models is driven by software requirements.

## II. SOFTWARE DEVELOPMENT PHASES

A software process is combination of well defined development phases. The prominent among them are software analysis, software design, and software testing and maintenance activities [3].

### A. Software requirement specification

After feasibility study, the important task is to prepare a document that refers to SRS document. In the requirement analysis, the development team leader conducts from informal to formal meeting with the client to better understand user's requirement [4]. Here the attention is to understand every requirement details of the prospective customers of the software. It is refined further and a series of meeting takes place to ascertain the scope of software and if that actually addresses the user's requirements.

### B. Technical Analysis of Software

After requirement gathering and requirement elicitation activities, the next technical activity is software requirement analysis. Under this, a detailed analysis is done to understand how software would behave to requirements [5]. Data flow diagram, Behavior diagrams, Entity relationship diagrams are important part of structured analysis. A complete UML diagrams are drawn to understand the detailed object oriented diagrams. There are many supplementary diagrams like process specification, control specification, data object description etc. These diagram further clarify the description of the diagram are helpful to implementation team members of the software.

### C. Software Design

The design of software is the backbone of success of software. The design description is built on data design, architecture design, interface design and the procedural or the modular design. The

design is aimed addressing the risk factors, technological developments with stakeholders. Some important issues like time and budget constraints need special attention[6]. The software design approach facilitates the maintenance activities which are so specific to software product.

#### *D. Implementation*

This phase of software development is dedicated to software implementation. The SRS document is detailed about user requirement and its implementation is done in implementation phase. Usually at the back end the database is created to store and fetch data. At the front end object oriented languages are used to address the design of software.

#### *E. Testing*

Software testing activities are done to ascertain that the built software performs as per the SRS document [7]. Bugs, if any, are traced to requirement specifications and a set of testing activities are carried out to remove errors. A substantial amount and efforts are invested in software testing activities.

#### *F. Deployment and Maintenance*

The next important software development activity is deployment. The software is deployed as per client needs and a feedback mechanism is put in place to encourage feedback from users. Such feedback is essential for long term maintenance and proper functioning of the software [7]. Software testing team is responsible for any regression testing of the software. There may be changes in software requirement and is well attended by testing team. Software configuration management activities are in place to address the version control of the software.

### III. SOFTWARE DEVELOPMENT PARADIGM

#### *A. The linear sequential model*

One of the basic and traditional model is the linear sequential model or the classic life cycle model. It is one of the oldest

model and in fact often used concept. The water fall name is synonymous with the fact that the model proceeds in a linear manner. In this every phase should be completed before next phase starts. The success of the model depends on the completion of every phase of the model. The model is best suited when the development team has experience on similar projects. Because in that case, the team can easily carry out system analysis, system designs, system implementation and system maintenance activities [5]. However, there are drawbacks in this model and the biggest is this model proceeds linearly in forward direction. Here, if some problem has occurred in analysis or design then it will only move forward. So to fix the problem, a development team has to start all over again. Also this model requires that all requirements of the software to be built should be known beforehand. This is absolutely essential.

#### *B. V&V Model*

Verification and validation model is a model ideally suited for software testing. The idea behind this model is as the software is developed, the testing techniques may be simultaneously applied. The various software development like analysis, design, coding is linked with testing strategies. The V & V model says white box testing and black box testing may go together with development. In this model, we do not deviate from the goal of the project because of each stage of testing.

It is Easy to understand and implement. It is suitable for early error removal and has high level of success as compared to a waterfall because of each testing phase. It is Inflexible and tough model with High risk associated with this model. The goal is not clear on this model.

#### *C. Iterative Model*

Using an iterative model, a project can be developed in small portions, each of which

contains some additional functions. In this model, there is no need for a complete requirement, unlike the V-shaped model and the Waterfall model, before developing software. With each iteration, some additional requirements. An updated version of the software is added and created, and this process continues until a complete project is developed. One of the advantages of an iterative model over another SDLC. The methodology is that we get a working version of the application in the early stages of the process, and therefore the introduction of changes is cheaper. One of the drawbacks is that resources can be eaten quickly by repeating the process over and over.

#### *D. Spiral Model*

The spiral model was built by Boehm in 1981[8-12]. It is by far the most suitable model for real time systems. The software development tasks are divided among six to eight components. At the end of each iteration one useful module is delivered. Next the requirement is revisited having consultations with the customer and a refine takes place at every possible way. In such away a system gets evolved. The spiral model supports the idea that a systemic iteration of steps is the right way to implement any software. The spiral model is divided into tasks like customer communication, planning, risk analysis, engineering, construction and release and customer evaluations. In the engineering phase the technical analysis takes place and also the design is tasked into. In the construction and release phase, the system is implemented as well as a through testing place. By far this is the most applicable to present system of application of software development.

#### REFERENCES

- [1] N. M. A. Munassar and A. Govardhan, "A comparison between five models of software engineering," *Int. J. Comput. Sci. Issues*, vol. 7, no. 5, pp. 95–101, 2010.
- [2] S. U. Khan, M. Niazi, and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: A systematic literature review," in *Proc. 4th IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Limerick, Ireland, Jul. 2009, pp. 207–216.
- [3] J. E. Purcell, "Comparison of software development lifecycle methodologies," in *Proc. Web Appl. Secur. Workshop*, Feb. 2007, pp. 1–9.
- [4] I. Sommerville, *Software Engineering*, 7th ed. Reading, MA, USA: Addison-Wesley, 2004.
- [5] H. D. Benington, "Production of large computer programs," in *Proc. Navy Symp. Adv. Programm. Methods Digit. Comput. Office Naval Res.*, Washington, DC, USA, 1956, pp. 15–28.
- [6] W. W. Royce, "Managing the development of large software systems," in *Proc. IEEE Wescon*, Aug. 1970, pp. 328–338.
- [7] B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2011.
- [8] B. W. Boehm, *Software Engineering Economics*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1981.
- [9] V. R. Basili and A. J. Turner, "Iterative enhancement: A practical technique for software development," *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 4, pp. 390–396, Dec. 1975.
- [10] C. Larman and V. R. Basili, "Iterative and incremental developments. A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.
- [11] P. Kruchten, *The Rational Unified Process—An Introduction*. Reading, MA, USA: Addison-Wesley, 2000.
- [12] Sanjay Chouhan, Vivek Singh Kushwah, Debendra Kumar Panda, "Comparison of simple rectangular and slotted ground micro-strip patch antenna for WLAN" *Engineering and Technology Journal for Research and Innovation*, ISSN 2581-8678, Vol. 1, Issue 1, pp.20-27, Dec 2018.