International Research Journal of Multidisciplinary Science & Technology
www.irjmrs.com ISSN : 2455-930X

Software development lifecycle and a review of development models

Deepshikha Jamwal[1] & Devanand[2]

[1]*Govt. Degree College Boys Udhampur, University of Jammu*
[2]*Department of Computer Science, Central University Jammu*

---***---

Abstract - *Software development life cycle helps developer to select a strategy to develop the software. A software development paradigm has its own set of tools, methods and procedures, which are expressed clearly and defines software development life cycle. The focus here about Software development life cycle. So, due to that different types of projects have different requirements, therefore, it may be required to choose the SDLC phases according to the specific needs of the project. These different requirements and needs give us various software development approaches to choose from during software implementation.*

***Key Words*:** software development, software development lifecycle, software management process.

1.INTRODUCTION

Software development life cycle (SDLC) is a method by which the software can be developed in a systematic manner and which will increase the probability of completing the software project within the time deadline and maintaining the quality of the software product as per the standard. The System Development Life Cycle framework provides a sequence of activities for system designers and developers to follow for developing software. It is often considered as a subset of system development life cycle. Any software development process is divided into several logical stages that allow a software development company to organize its work efficiently in order to build a software product of the required functionality within a specific time frame and budget. In this paper we focus on the time covering the development of one particular version of a software product or system, from requirements definition until requirements validation. We believe we will give a more clear view of software development lifecycles than when models are presented unrelated[1].

1.1.Software Process Model

A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective as:

1. Specification.
2. Design.
3. Validation.
4. Evolution.

General Software Process Models are:

1. Waterfall model: Separate and distinct phases of specification and development.
2. Prototype model.
3. Rapid application development model (RAD).
4. Evolutionary development: Specification, development and validation are interleaved.
5. Incremental model.
6. Iterative model.
7. Spiral model.
8. Component-based software engineering: The system is assembled from existing components.

There are many variants of these models e.g. formal development where a waterfall-like process is used, but the specification is formal that is refined through several stages to an implementable design [2, 3].

1.2.The Waterfall Model

The waterfall model is believed to have been the first process model which was introduced and widely followed in software engineering. The innovation was that the first time software engineering was divided into separate phases. In the early 1970's there was no awareness of splitting up software development into different phases.

1.3.The V Model:

A further development of the waterfall model led to the so called "V-Model". The individual steps of the process are almost the same as in the waterfall model. However, there is one big difference. Instead of going down the waterfall in a linear way the process steps are bent upwards at the coding phase, to form the typical V shape [4].

1.4.The Prototype Model:

This prototype is developed based on the currently known requirements. By using this prototype, the client can get an "actual feel" of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.

1.5.The RAD model:

In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

1.6.The Iterative model:

It is developed to overcome the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system [5].

1.7.The Spiral Model:

The Spiral Model is the most flexible and agile of all traditional software process models. The process begins at the centre position. From there it moves clockwise in traversals. Each traversal of the spiral usually results in a deliverable. It is not clearly defined what this deliverable is.

2.THE SOFTWARE DEVELOPMENT PROCESS

After selecting a Process Model for your business it is necessary to utilize it. This first means to define the overall Product Life Cycle with its engineering disciplines and phases. The next step would be to define detailed activities within each of the phases. The documents and other work products that shall be an outcome of the activities have to be defined in the next step [6].

3.SOFTWARE CHARACTERISTICS

The software has very special characteristics e.g., "it does not wear out". Its behaviour and nature is quite different from other products of human life.

Some of the important characteristics are discussed below:

1. Software does not wear out.
2. Software is flexible.
3. Software is not manufactured.
4. Reusability of components[6].

4.SOFTWARE APPLICATIONS

Software has become integral part of most of the fields of human life. Software applications are grouped in to eight areas for conveniences and these are:

System Software. Infrastructure software comes under this category like compliers, operating systems, editors, drivers, etc.

Real Time Software. These software are used to monitor, control and analyze real events as they occur. E.g. weather forecasting etc.

Embedded Software. This type of software is placed in "Read- Only-Memory (ROM)" of the product and controls the various functions of the product.

Business Software. The software designed to process business applications is called business software. Business software could be payroll, file monitoring system, employee management, and account management etc

Personal Computer Software. The software which comes under this category is word processors, computer graphics, multimedia and animating tools, database management, computer games etc [7].

Artificial Intelligence Software. Artificial Intelligence Software makes use of non-numerical algorithms to solve complex problems that are not amenable to computation. e.g expert system, artificial neural network. Signal processing software etc.

Web based Software. Software like CGI, HTML, Java, DHTML etc.

Engineering and Scientific Software. Scientific and engineering application software is grouped in this category. Huge computing is normally required to process data. e.g. CAD/CAM packages, SPSS, MATLAB, Engineering Pro, Circuit analyzers etc [8].

5.FRAMEWORK MODEL

The model tries to simulate the development of information system within for small business organizations. The model recreates these development considering two different development methodologies:

The first is a **Sequential waterfall-based** approach & the second is an **Iterative-based** approach that gathers elements from agile and extreme programming methodologies.

The model defines project static set of tasks to be worked through different development phases. Each phase has a different development rate. In the sequential approach the first three phases have two distinctive parts: **Development and Verification**.

During development approach the development tasks are being worked out but without their verification. However, verification starts after all tasks have been worked out. During this part tasks that need to be reworked out are detected and sent back and the rest of the tasks are approved and sent forward to the next phase. The next phase doesn't start until all the tasks have been verified and approved.

However, because of the lack of insight and verification right at the initial stages of a project, a subset of tasks where reworking is not identified during verification is mistakenly approved. In the next phase these tasks will generate more reworking. In the last phase of testing, the verification accuracy increases and most of these tasks are finally identified and sent back to be fixed.

In the iterative approach, analysis, design, and coding are grouped into one big first phase that includes several small cycles of validation. The size of each cycle can change between methodologies. A spiral model could have iterations of many weeks. Some more recent approaches such as extreme programming, recommend performing integration and validation tasks on daily basis.

6.PROPOSED RECOMMENDATIONS

Based on the above study, some recommendations are being put forth which are enumerated below:

RECOMMENDATION 1: THE Time and information flows in a model should be separated and separate information flow and time flow models should be defined for the life cycle.

RECOMMENDATION 2: USE activities as the basic ingredients for grouping of different objects in the process to perform different tasks and this grouping should be partly driven on the basis of their basic characteristics.

RECOMMENDATION 3: Use V-process model to sequence different groups of activities to represent information flow. The selection for the V-process model is to give importance to information flow order rather than timing on the workflows in various perspectives like whether the activity is decomposition or compositional in nature. The corresponding decomposition/ composition steps need information transfer, which can be easily depicted when they are at opposite positions in the V process model.

Furthermore, there are few more ideas available from other approaches, which can also be employed. The inputs of the activities and their outputs if defined properly for each phase can generate some more ideas. To make explicit what is used and produced in these activities, input and output artifacts are defined for each phase. These come, both from the previous phases as well as from the outside [9].

7.CONCLUSION

A study is given about different models. There are a number of development models exist. This paper explained five different models out of those. First one is waterfall model which provide base for the other development models. Then its enhanced models are explained. Iterative model, Spiral model, V shaped model, RAD and prototype model.

8.FUTURE SCOPE

The work presented in this paper is the survey of software development models. The software community is today too immature to be able to come to a consensus about how the lifecycle of software development should be described. There are lot of limitations in existing models. In future lot of work can be done to produce a new software development processes and models.

REFERENCES

[1] Apoorva Mishra, Deepty Dubey, "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios", Volume 1, Issue 5, ISSN: 2321-7782 (Online), International Journal of Advance Research in Computer Science and Management Studies, October 2013.

[2] 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.

[3] Laura C. Rodriguez Martinez, Manuel Mora ,Francisco,J. Alvarez, "A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles", Proceedings of the 2009 Mexican International Conference on Computer Science IEEE Computer Society Washington, DC, USA, 2009

[4] Philippe Kruchten, The Rational Unified Process: An Introduction, Second Edition, ISBN 0-201-70710-1, Addison-Wesley, 2000.

[5] Steve McConnell, Rapid Development, Taming Wild Software Schedules, ISBN 1-55615-900-5, Microsoft Press, 1996.

[6] Jennifer Stapleton, DSDM – Dynamic Systems Development Method, ISBN 0-201-17889-3, Pearson Education, Harlow 1997.

[7] Deepshikha, Devanand, "Analysis of Software Development terms and user satisfaction level", INDIACOM, ISBBN 093-7529, ISBN 978-904526-6-3, 2009.

[8] Deepshikha Jamwal & Devanand**,** "Comparative Study of Different Software Development Models For Small Organizations", ISBN: 978-981-08-2465-5, IEEE 2009.