ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2019; 5(2): 77-80
www.allresearchjournal.com
Received: 10-01-2019
Accepted: 12-02-2019

Gagandeep Kaur
Assistant Professor, Guru
Nanak College, Ferozepur,
Punjab, India

Correspondence Author:
Gagandeep Kaur
Assistant Professor, Guru
Nanak College, Ferozepur,
Punjab, India

A review on SDLC (Software development life cycle)

Gagandeep Kaur

Abstract
SDLC stands for Software Development Life Cycle. SDLC is a process that consists of a series of planned conditioning to develop or alter the Software Products. The present study will give you an overview of the SDLC basics, SDLC models available and their employment in the assiduity.

Software project development is one of the most powerful, vital and important issue in the world of computers. It is the subset of system development life cycle (SDLC). SDLC is all about the minimization of risk and failure and maximization of quality of the software product. The software development process is the very complex thing without any proper step by step generating procedure so to make the software development processes simple and systematic the software development life cycle came in to existence. This is the systematic and structural method of software developing process. The SDLC defines the framework that includes different activities and tasks to be carried out during the software development process. There are various software development life cycle models that are used in the software development process heaving their own advantages and disadvantages in this paper we have included five of these SDLC models like waterfall model, V shaped model, prototype model, spiral model and iterative model for comparison of existing models.

Keywords: software development life cycle, models, prototype, modeling, development, risk analysis and comparative analysis

Introduction

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

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.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- 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.

The software system planning superintendency is the clear need in now a day. A plan is like a road chart. Software engineering is a discipline for developing high quality software for computer rested systems. Software engineering is all about the quality focus, process, ways and tools that are used in software development process. The software development process defines a structure that includes different exertion and tasks. We can say the process defines what different conditioning and tasks are to be carried out during software development. (Or manner of developing the software). The software planning direction provides the specialized way starting how to apply the software manners that include Communication, Must Analysis, Analysis and Design modeling, Program construction, Testing and support. Software development life cycle SDLC is the neat approach to complete the software development process within the time and maintain quality of the software. System development life cycle provides the set of conditioning to be carried out during the system development and it's hourly called that software development life cycle. Software development is divided into set of conditioning that allow any software development company to control the software product freely. The software development life cycle models use the step by step approach to complete the software development process. However, the end product will also be strong and game can get success, If the process is strong. While developing the good software product the devisers that are directly or edgeways included in this process should keep the following points in their mind.

- A quality focus
- Process
- Methods – Tools

The software process model is the representation of process it presents the description of a process as.

- Specification
- Design
- Validation
- Evolution

The software development life cycle is all about.

- Understanding the problem. i.e. (problem domain)
- Decide a plan for solution. i.e. (solution domain)
- Coding the planed solution
- Test the actual program
- Maintain the product

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models. Each process model follows a Series of steps unique to its type to ensure success in the process of software development.
Following are the most important and popular SDLC models followed in the industry -

- Waterfall Model
- Iterative Model
- Spiral Model
- V- Model
- Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

The Waterfall Model

This model was proposed by Royce in 1970. This is the classical model of the software engineering. The model is one of the oldest models used in the government programs and in multitudinous of the major companies. The model is also called as the straightaway successional model or classic life cycle. This is the straightaway successional SDLC model followed by the following phases Needful Analysis, Design, Perpetration, Testing and Preserving. The phases are fixed in such a manner the phase once can’t be repeated again. As the water fall model is the classical model so it serves as the birth of all other models. The cascade model consists of several non-lapping stages as shown in the figure bellow. It's one way avenue with no turning back. You cannot move backward once phase “X” is over, you can move to phase “Y”. There's no provision of moving backward.

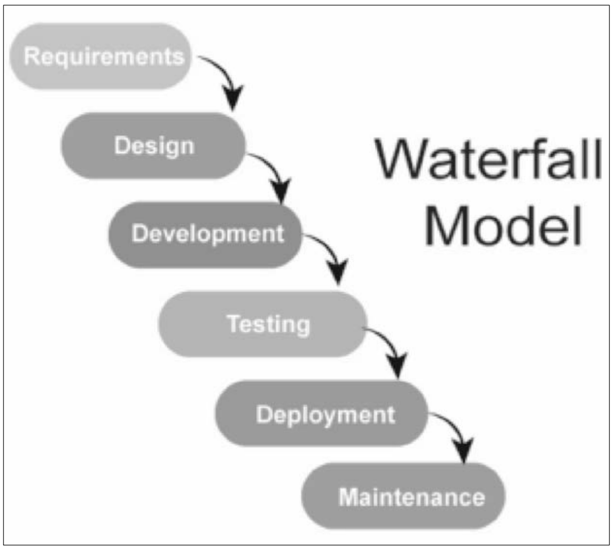

Fig: Waterfall Model

Requirement Analysis

The software development process starts with the communication between the contriver and the account. According to cascade model the account must state all the requirements at the startup of the project

- **System and Software design:** system and software design includes the complete estimation like cost, time, manpower and scheduling. The complete timeline map for the blueprint development and chasing.
- **Application and unit testing:** It includes the requirements and the program design from developer’s side.
- **Integration and system testing:** this phase deals with whether the software meets the specified demands given by the user.
- **Operations and Maintenance:** After software release this phase is used for advance of problems.

Advantages

- Easy to understand.
- Define before design.
- It's used for program with simple and strict deadlines.

Disadvantages

- Idealized.
- Problems remain uncovered until testing.
- Demands aren't clear.
- Late delivery.
- Not easy to integrate risk management
- Tricksy to make any changes.

The Iterative Waterfall Model

The problems with the classical fall model created a demand of new model. The iterative model came into existence in order to manage with the problems of the original waterfall model the iterative fall model is enhanced rendition of classical fall model which could supply faster results warrant subordinate time and heaving good suppleness. In iterative model program (project) is divided into small parts and this allows developer squad to go freely and fast for their target and secure their ultra-expensive feedback from users. The project that's divided into small parts presently each part is actually a mini waterfall process.

Advantages

- Much better model of software process.
- User can get Feedback.
- Used in that type of projects were must-haves aren't clear.
- Document driven process.
- Works well on week teams.

Disadvantages

- Not easy to manage.
- Not clear mile stones
- No stage is finished really.

The Prototyping Model

Prototype model is the evolutionary process model. Evolutionary Process models are the iterative type models using this model the contriver cam develop inchmeal more complete account of software. In prototype model presently we've the quick design phase through which the formulator has to fast design the software after necessary gathering.

Advantages

- Early visibility
- Higher outputs
- Cost reductive
- Account can actually feel with the system, i.e., feedback
- Early design

Disadvantages

- Possibility of unfinished systems
- Possibility of inadequate system
- Lack of suppleness.
- Not suitable for large systems.
- Management is really complicated.

Incremental Model in SDLC

The incremental model is not a separate model. It is essentially a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC life cycle process is repeated, with each release adding more functionality until all requirements are met. In this method, every cycle act as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

The Corkscrew Model

The corkscrew model was developed on the end of the 1980s, it was outlined by Barry Boehm, and introduces existent that other models did not take into account, which is hazard analysis. In essentiality, the coiling model attempts to bring together critical aspects of some other prominent models (namely the cascade, incremental, and evolutionary prototyping), in an attempt to gather the most becoming traits from each one, because specific blueprints might be more or less adaptable to specific models. It has lots of all above said models. It's also called Meta model.

Advantages

- New prototype acquired every time
- Employ capabilities
- Better productivity
- It has methodical accumulative approach
- Elimination of misdoings in early stage

Disadvantages

- Not proper cost and time estimation
- Cost of risk analysis is high on large programs
- Complex

The V - Shaped Model

This model can be considered as extension of waterfall model. In waterfall model we move in a straight way while in V shaped model process route are guileful overhead the cracking phases to form typical V Shape. The relationship between each phase of the development process and its associated phase of testing. As testing is one of the important part of Software development the V shaped model emphasis is more on testing. The V shaped Model is fair same as waterfall model both the models are back-to-back. Requisites have to be really clear before the strategy starts because it's really ultra-expensive to go back to requisites and make changes for the subsisting strategies.

Advantages

- High volume of risk analysis
- Good for critical strategies
- Early production
- Easy to manage due to strictness of model
- Easy to understand

Disadvantages

- Not good model for object acquainted programs
- Not good for long and ongoing programs
- Not suitable were essentials have high imminence of changing
- Can be extravagant model.
- Don’t work well for small programs

Conclusion

The Software Development Life Cycle (SDLC) is a systematic process for building software that ensures the quality and correctness of the software built. After completing the anatomizing and comparison we conclude that the models are developed from 1970 to 1999. Every model has its own advantages and disadvantages and every model came into actuality to do with the problems of being model of that time. In now a day’s waterfall and spiral are the most ordinarily used in the software development process and the other models are used according to the conditions of the software products. The software contrivers use the models according to the size of the software that's to be developed. All the developers and users look at the low coat, risk, high quality and small cycle of time, so that the productivity and quality of the software product can optimized.

References

1. Aggarwal KK, Yogesh Singh. Software Engineering 3rd Edition.
2. Karlm. Software Lifecycle Models', KTH 2006.
3. Vanshika Rastogi. Software Development Life Cycle Models-Comparison, Consequences. ISNN: 0975-9646 International Journal of Computer Science and Information Technologies 2015;6(1):168-172.

4. Sanjana Taya. Comparative Analysis of Software Development Life. ISSN: 2229-4333 (print), ISSN: 0976-8491(online) 2011, 2(4).
5. Nabil Mohammed Ali Munassar, Govardhan A. A Comparison between Five Models of Software Engineering Ijcsi International Journal of Computer Science 2010, 7(5).
6. Comparative Analysis of Different types of Models in Software Development Life Cycle-Ms. Shikha Maheshwari, Prof. Dinesh Ch. Jain
7. Kuhl JJ. Project Lifecycle Models: How they differ and when to use those 2002. www.businessesolutions.com.
8. Rlewallen. Software Development Life Cycle Models 2005. http://codebeter.com.
9. Barry Boehm. Spiral Development: Experience, Principles, and Refinements, edited by Wilfred J Hansen 2000.
10. Roger Pressman. Software Engineering: A Practitioner’s Approach, Sixth Edition, McGraw-Hill Publication.