**ABSTRACT**

The software development life cycle (SDLC) is used to design, develop and produce high quality, reliable, cost effective and within time software products in the software industry. This is also called software development process model. There are different SDLC process models are available. In this paper I have tried to describe different SDLC models according to their best use. There are many papers which have written in this regard. I will also use their knowledge or findings in this paper. The main purpose of this paper is to explain some of important SDLC models like Waterfall Model, Iterative Model, Spiral Model, V-Model, Big Bang Model, Agile Model, Rapid Application Development Model and Software Prototype. The main purpose of this paper is to explain advantages and disadvantages of these SDLC models. I will also describe which SDLC model is best fit for which type of software applications.

# 

|  |  |  |
| --- | --- | --- |
| **S NO** | **INDEX**  **CONTENTS** | **PAGE NO** |
| 1 | Introduction | 1 |
| 2 | History | 2 |
| 3 | What is SDLC | 3 |
| 4 | Phases of SDLC | 4 |
| 5 | Models of SDLC | 7 |
| 6  7 | 1. Waterfall Model 2. Incremental Model 3. Evolutionary Model 4. Spiral Model 5. RAD (Rapid Application Development) 6. Extreme/Agile Development   Conclusion  Reference | 7  9  10  11  12  13  14  15 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 1. INTRODUCTION

SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality software’s. 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.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.

# 

Page 1

# 2. HISTORY

The Systems Life Cycle (SLC) is a type of methodology used to describe the process for building information systems, intended to develop information systems in a very deliberate, structured and methodical way, reiterating each stage of the life cycle. The systems development life cycle, according to Elliott & Strachan & Radford (2004), "originated in the 1960s,to develop large scale functional business systems in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines".

Several systems development frameworks have been partly based on SDLC, such as the Structured Systems Analysis and Design Method (SSADM) produced for the UK government Office of Government Commerce in the 1980s. Ever since, according to Elliott (2004), "the traditional life cycle approaches to systems development have been increasingly replaced with alternative approaches and frameworks, which attempted to overcome some of the inherent deficiencies of the traditional SDLC

Page 2

# 3. WHAT IS SDLC

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.

Software Development Life Cycle consists of details steps and activities which describes how to design, develop, maintain, replace, alter, enhance, test or even launch a software.

The activities can be broken down into a very detail level but at the same time they can be grouped into five core categories.

Page 3

# 4. PHASES OF SDLC

# Software Development Life Cycle phases:

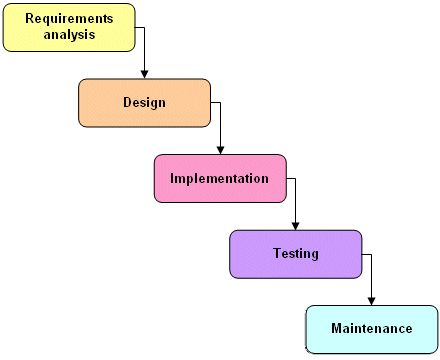
1. Requirement gathering and analysis

2. Design

3. Implementation or coding

4. Testing

5. Maintenance



Page 4

**1) Requirement gathering and analysis:**

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) 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.

Page 5

**3) Implementation / Coding:**

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 software development life cycle.

**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 :**

Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

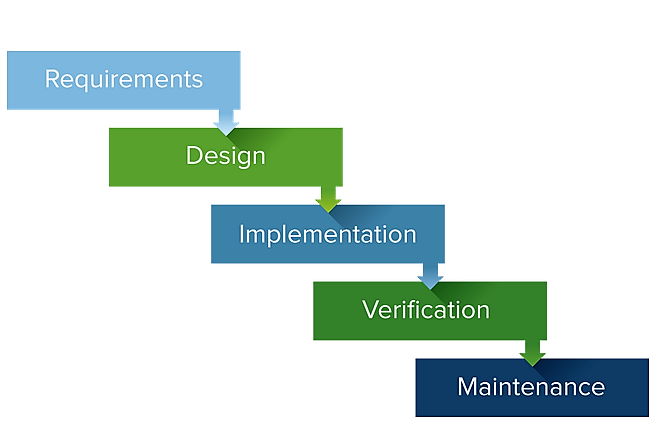
Page 6

# 5. MODELS OF SDLC

**Software Development Life Cycle Models :**

1. Waterfall Model
2. Incremental Model
3. Evolutionary Model
4. Spiral Model
5. RAD (Rapid Application Development)
6. Extreme/Agile Development
7. **Waterfall Model :**

Waterfall Model is the earliest, best-known and most commonly used methodology. It is a sequential life cycle that is simple to understand and use. Each phase has to be completely finished before another start which means no overlapping is allowed. The output of each phase serves as the input for the next stage.



Page 7

# Advantages :

* Simple goal.
* Simple to understand and use.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are Well documented.
* Easy to manage.
* Works well when quality is more important then cost/schedule.
* Customers/End users already know about it.

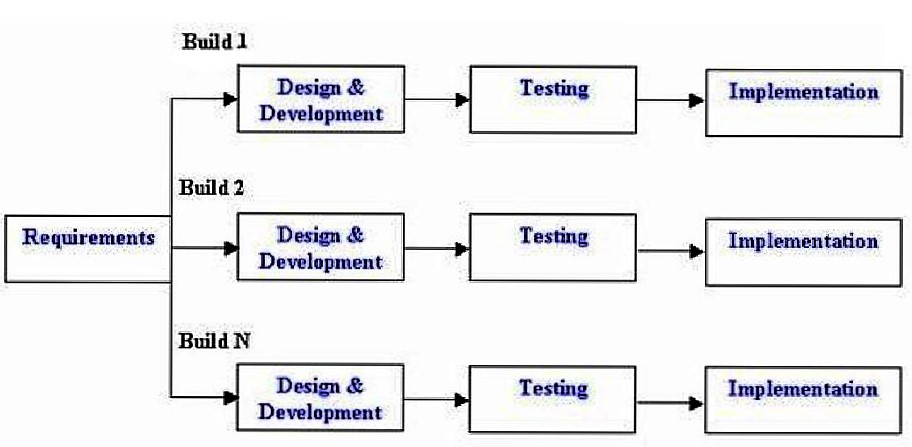
# Disadvantages :

* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.
* No working software is produced until late in the life cycle.
* Risk and uncertainty is high with this process model.
* Adjusting scope during the life cycle can end a project.
* Not suitable for complex projects.
* Users can only judge quality at the end.
* Attempt to go back 2 or more phases is very costly.
* Very risky, since one process can not start before finishing the other.

Page 8

# Incremental Model :

# Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.



# Advantages :

* Working functionality can be developed quickly in the life cycle.
* Results are obtained early and periodically.
* Parallel development can be planned.
* Progress can be measured.
* Less costly to change the scope/requirements.
* Testing and debugging during smaller iteration is easy.
* Easier to manage risk - High risk part is done first.

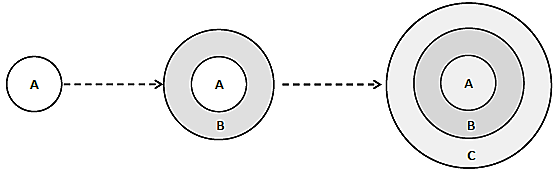
# Disadvantages :

* More resources may be required.
* Although cost of change is lesser but it is not very suitable for changing requirements.
* More management attention is required.
* Each phase of an iteration is rigid with no overlaps.

Page 9

1. **Evolutionary Model :**

Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.

****

# Advantages :

* Risk analysis is better.
* It supports changing requirements.
* Initial Operating time is less.
* Better suited for large and mission-critical projects.
* During life cycle software is produced early which facilitates customer evaluation and feedback.

**Disadvantages :**

* Not suitable for smaller projects.
* Management complexity is more.
* End of project may not be known which a risk is.
* Can be costly to use.
* Highly skilled resources are required for risk analysis.
* Project’s progress is highly dependent upon the risk analysis phase.

Page 10

1. **Spiral Model :**

This Spiral model is a combination of iterative development  process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

****

**Advantages :**

* Changing requirements can be accommodated.
* Allows for extensive use of prototypes.
* Requirements can be captured more accurately.
* Users see the system early.
* Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.

**Disadvantages :**

* Management is more complex.
* End of project may not be known early.
* Not suitable for small or low risk projects.
* Process is complex.
* Spiral may go indefinitely.

Page 11

# RAD (Rapid Application Development) :

# The RAD (RapidApplication Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

# rad.png

# Advantages :

# Time to deliver is less.

# Changing requirements can be accommodated.

# Progress can be measured.

# Cycle time can be short with use of powerful RAD tools.

# Productivity with fewer people in short time.

# Use of tools and frameworks.

# Disadvantages :

# 

# Management complexity is more.

# Resource requirements may be more.

# Suitable for systems that are component based and scalable.

# Suitable only when requirements are well known.

# Requires user involvement throughout the life cycle.

# Suitable for project requiring shorter development times.

# 

Page 12

# Extreme/Agile Development :

# 

# Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

# agi.png

# Advantages :

# 

# Promotes teamwork and cross training.

# Functionality can be developed rapidly and demonstrated.

# Resource requirements are minimum.

# Suitable for fixed or changing requirements

# Delivers early partial working solutions.

# Good model for environments that change steadily.

# Minimal rules, documentation easily employed.

# 

# Disadvantages :

# 

# Not suitable for handling complex dependencies.

# More risk of sustainability, maintainability and extensibility.

# An overall plan, an agile leader and agile PM practice is a must without which it will not work.

# Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.

Page 13

# 6. CONCLUSION

Software Development Life Cycle (SDLC) is the process of developing information systems through analysis, planning, design, implementation, integration maintenance and testing of software applications. SDLC is also known as information systems development or application development. The development of quality software involves the usage of a structured approach towards the design and development of the end product.

In a nutshell, the success of the SDLC process for building a successful software system rests on the following:

* Scope Restriction
* Progressive Enhancement
* Pre-defined Structure
* Incremental Planning at each of the stages

If each of these steps can be followed in its entirety, most of the risks that evolve in the software development life cycle can be mitigated. This article has provided a comprehensive explanation of the important and widely used Software Development Life Cycle (SDLC) phases and its models. It has also provided the advantages and disadvantages of each of these models.

Page 14

# 7. REFERENCE

* Gallaugher, J. (2012). Understanding Software: A Primer For Managers. *Information Systems: A Manager's Guide to  Harnessing  Technology* (1.4 ed.). Irvington, NY, United States: Flatworld Knowledge, Inc.
* Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software Development Life Cycle AGILE vs Traditional Approaches.  In *2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT* (Vol. 37).
* McMurtrey, M. (2013). A Case Study of the Application of the Systems Development Life Cycle (SDLC) in 21st Century  Health  Care: Something Old, Something New?. *Journal of the Southern Association for Information Systems*,*1*(1).
* Website: Systems Development Life Cycle. (2013). Wikipedia. Retrieved on July 12, 2013 from <http://en.wikipedia.org/wiki/Systems_development_life-cycle>
* Website: The Systems Development Life Cycle Basics. (n.d.). McGraw Hill Connect. Retrieved on July 12, 2013 from <http://highered.mcgraw-hill.com/sites/dl/free/0073376736/596593/AppendixD.pdf>

Page 15

Page 15