

COMPARISION OF DIFFERENT SDLC MODELS

Radhika D. Amlani
Research Scholar
Saurashtra University
Rajkot(Gujarat)

ABSTRACT

Software engineering is the area which is constantly growing. It is very interesting subject to learn as all the software development industry based on this specified area. Now a day, there are lots of software development life cycle models available. According to the requirements, software industry people are using it as per their requirements. As there are lots of SDLC models, they are used according to their requirements. So, it is needed to know the requirements in which the SDLC models is used. This paper is about to give information of the comparative study of different SDLC models.

General Terms

SDLC models, Waterfall model, Spiral model, RAD, RUP, Prototype model,

Keywords

SDLC Phase, Comparison of different SDLC models, Analysis of different SDLC models

1. INTRODUCTION

Software Development Life Cycle Model is used as a process of creating and altering current existing system. SDLC used in information system, systems engineering, and software engineering. SDLC can be thought of as a concept that used by many software development methodologies, which are currently available in market or software industry. SDLC provide a framework to create, plan and control any information system to be developed.

Developers worldwide will agree that building software takes more than just writing complex codes and implementing them in an environment. Developers usually start out their career in programming by developing programs or software according to their own plan and hope that someone will appreciate it. But once the developer is associated with a business or another software company, the creativity is limited to business and consumer needs. The pressure in creating accurate and efficient software is even bigger in the entrepreneurial stage.

To ensure developers have come up with the right software for the specific need, programmers have created steps on how a program could specifically created. This will ensure everything is built according to plan and tested extensively

before it could be implemented for public or formal use in business. Under these circumstances the term “Systems Development Life Cycle” was born. The need to create accurate and efficient software has led to the formalization on certain stages and phases on how a program should be built.

Simply put, SDLC or Systems Development Life Cycle is a series of steps observed by developers on building specific software. Developers follow certain steps to ensure they have the right software for the right demand.

The history of the term “Systems Development Life Cycle” is not very clear but it naturally came into being since the 1960s when developers started to create programs specific to a certain need. Slowly, the term has been observed by different software development companies. From a simple format of planning, building, testing and implementing, software companies have developed their own version of developing specific products for their clients. Each version of software development is called “Model”.

SDLC describes a process, which is used by engineers and developers to create and deploy all features or characteristic of information system. This features or characteristic describes requirements, validation, training and emphasizing ownership of the system. Whenever SDLC is used, the goal is to create a system that meets the primary objective of the owner within defined timeline and cost constraints. It also includes post installation stages like, deployment and maintenance, which has features like ease of use, installation of the software, for minimizing error.

There are so many SDLC based, software engineering models available in market now-a-days. Depending upon the suitability, the software engineering model can be used to put forward any software project. Each of the methodologies or models has different level of risk and benefits to cope with the project requirements, budget and estimated completion timeline. There are models which are suitable for large project, where some focus on lightweight process that allow rapid changes throughout whole software development life cycle. Below is the comparison of the different SDLC models which could help the developer to choose the proper SDLC model to implement their project to develop.

Figure- I Comparison of SDLC Models

No	Model/features	Waterfall Model	Prototype Model	Spiral Model	Iterative and Incremental model
1	Specification of All the Requirements in the beginning	Yes	Not all and Frequently Changed	Not all and Frequently Changed	Not all and Frequently Changed
2	Project Cost	Almost as Estimated	Above Estimated Cost	Very Costly	Above Estimated

					Cost
3	Guarantee of Success	Low	Moderate	High	High
4	Required Expertise	Moderate	Moderate	High	Moderate
5	Overlapping Phases	No	Yes	No	Yes as Parallel development is there
6	Process	Heavyweight Process	Light weight Process	Heavyweight Process	Light weight Process
7	Framework type	Linear	Iterative	combination of Linear and iterative	Combination of Linear and Iterative
8	Rework cost	High	Not Low	High	Almost High
9	Testing	After coding phase Completed	After every iterative prototype model	At the End of Engineering Phase	After Every Iteration
10	Customer Involvement	Low	High, After Each Iteration	Low, After Each Iteration	High, After Each Iteration
11	Basic business Knowledge Required	Not much	Moderate	Not Much	Moderate
12	Suitable Project Size	Large Scale	Low to Medium Scale	Large Scale and Complex	Low to Medium Scale
13	Cost Control	Yes	No	Almost Yes	No
14	Simplicity	Simple	Moderate	Complex	Moderate
15	Risk Involvement	High	Low	Low	Low
16	Flexibility	Rigid	Much Flexible	Much Flexible	Much Flexible
17	Maintenance	Least Maintainable	Maintainable	Yes	Maintainable
18	Changes Incorporated	Difficult	Easily	Easily	Easily
19	Reusability	Least Possible	To some Extent	To some Extent	To some Extent
20	Documentation and Training	Necessary	Yes But Not Much	Yes	Yes But Not Much
21	Time Frame	Very Long	Long	Long	Long
22	Availability of Working Software	At the End of the Life Cycle	At the End of Every Iteration	At the End of Every Iteration	At the End of Every Iteration
23	Customized product	Least Possible	Possible	Possible	Much Possible
24	Customer Control over Administrator	Very Low	Yes	Yes	Yes
25	Required Team Creativity	No	Yes But Not Much	Yes But Not Much	Yes
26	Knowledge Transfer	No	Yes But Not Much	Yes But Not Much	Yes
27	Team size	Large Team	Small Team	Large Team	Not Large Team
28	Primary Objective	High Assurance	Rapid Development	High Assurance	Rapid Development
29	Implementation	Easy	Easy	Complex	Easy
30	Release Cycle	Big band(All Functionality at Once)	In Phases	Big band(All Functionality at Once)	In Phases

Figure- II Comparison of SDLC Models

No	Model/features	V-shaped Model	RAD Model	RUP Model	JAD Model
1	Specification of All the Requirements in the beginning	Yes	Not all and Frequently Changed	Yes	Not all and Frequently Changed
2	Project Cost	Almost as Estimated	Almost as Estimated	Almost as Estimated	Expensive
3	Guarantee of Success	Moderate	Very Good	Very High	High
4	Required Expertise	Moderate	Moderate	High	Very High
5	Overlapping Phases	No	Yes	No	Yes
6	Process	Heavyweight Process	Light weight Process	Heavyweight Process	Lightweight Process
7	Framework type	Non linear	Prototype and Iterative	Iterative and Incremental	Incremental
8	Rework cost	High	Not very High	High	High
9	Testing	After completion of Each Iteration	After Completion of Coding	At the Construction Phase	After Coding Phase
10	Customer Involvement	Low	High	High, After Each Iteration	Continuous
11	Basic business Knowledge Required	Not Much	Required	Required	Very Much
12	Suitable Project Size	Large Scale	Low to Medium Scale	Small as well as Large Scale	Small as well as Large Scale
13	Cost Control	Yes	No due to Urgent Requirement	No	Almost No
14	Simplicity	Moderate	Simple	Complex	Simple
15	Risk Involvement	Not High	Very Low	Low	Low
16	Flexibility	Little Flexible	Very Flexible	Very Flexible	Very Flexible
17	Maintenance	Little Maintainable	Easily Maintainable	Hard to Maintain	Maintainable
18	Changes Incorporated	Difficult	Easily	Easily	Easily
19	Reusability	Little Possibility	Yes	Yes	Yes
20	Documentation and Training	Yes	Limited	Very Limited	Yes
21	Time Frame	Long	Short	Long	Moderate
22	Availability of Working Software	At the End of the Life Cycle	At the End of the Life Cycle	At the End of the Life Cycle	At the End of Every Iteration
23	Customized product	Least Possible	Possible	Possible	Possible
24	Customer Control over Administrator	Low	Yes	Yes	Very Much
25	Required Team Creativity	No	yes	Yes	Yes
26	Knowledge Transfer	No	Yes	Yes	Yes
27	Team size	Small Team	Small Team	Large Team	Large Team
28	Primary Objective	High Assurance	Rapid Development	High Assurance	Rapid Development

29	Implementation	Easy	Easy	Complex	Moderate
30	Release Cycle	Big band(All Functionality at Once)	In Phases	Big band(All Functionality at Once)	In Phases

Figure- III Comparison of SDLC Models

No	Model/features	Scrum Model	Extreme Programming Model
1	Specification of All the Requirements in the beginning	Not all and Frequently Changed	Not all and Frequently Changed
2	Project Cost	Almost as Estimated	Almost as Estimated
3	Guarantee of Success	High	High
4	Required Expertise	Very High	Moderate
5	Overlapping Phases	Yes	Yes
6	Process	Light weight Process	Light weight Process
7	Framework type	iterative and incremental	Iterative
8	Rework cost	High	High
9	Testing	After Coding Phase	automated testing while coding
10	Customer Involvement	High	Continuous
11	Basic business Knowledge Required	Very Much	Very Much
12	Suitable Project Size	Large Scale	almost small scale
13	Cost Control	No	No
14	Simplicity	Simple	Simple
15	Risk Involvement	Not High	Not High
16	Flexibility	Flexible	Very Flexible
17	Maintenance	Maintainable	Easily Maintainable
18	Changes Incorporated	Easily	Easily
19	Reusability	Yes	Yes
20	Documentation and Training	Limited	Limited
21	Time Frame	Moderate	Short
22	Availability of Working Software	At the End of Every Iteration	At the End of Every Iteration
23	Customized product	Possible	Possible
24	Customer Control over Administrator	Yes	Very Much
25	Required Team Creativity	Yes	Yes
26	Knowledge Transfer	Yes	Yes
27	Team size	Large Team	Small Team
28	Primary Objective	Rapid Development	Rapid Development
29	Implementation	Moderate	Easy
30	Release Cycle	In Phases	In Phases

2. ACKNOWLEDGMENTS

I am thankful to Dr. Kishor Atkotiya Head, Department of Computer Science,

J. H. Bhalodia Women's College,

Rajkot - 360 002, INDIA

Who has guided me to analyze different SDLC.

3. REFERENCES

[1] www.en.wikipedia.org/wiki/Systems_development_life-cycle

[2]www.en.wikipedia.org/wiki/SDLC

[3]www.waterfall-model.com/sdlc/

[4]www.condor.depaul.edu/jpetlick/extra/394/Session2.ppt

[5]www.searchsoftwarequality.techtarget.com/.../systems-development

[6]<http://searchsoftwarequality.techtarget.com/answer/Software-development-life-cycle-phases-iterations-explained-step-by-step>

[7]www.allinterview.com › Categories › Software

[8] www.vtglobal.com/software-development-life-cycle.html

[9]www.e-digg.com/services/software/sdlc-life-cycle.html - United States

[10]www.shazsoftware.com/software-development-life-cycle.html