

EVERSION OF NEW SOFTWARE DEVELOPMENT TOOL (SDLC-14)

Ms. Sonam Arora
Department of Computer Science
Guru Nanak Khalsa College ,Karnal

Abstract: In the current environment there is a extreme level of competition in it industry for developing the proposed software. There are several software development lifecycle models and every model has the best benefits which lays special impact on development and unforgettable demerits also. In this research, emphases are lays on evolving new tool which worked as an software development model includes the best benefits of each model. It specifies stress emphasis on special elements of development such as client's satisfaction ,risk exposure and feasibility of project. Tool developed for maximizing the interaction between user and developer in order to have better communication.

Keywords: SDLC(software development life cycle),pmt(project management techniques),SDLC phases.

1. Introduction

The term “Software Engineering” coined by Margaret Hamilton was first used in 1968. Software engineering is a quantitative approach which specifies the technical and mental capability for developing the project and tests the patience of users. Software engineering is to engineer the project in logical point of view according to plighted client’s requirement. Software engineering is the control unit which controls and manage the overall aspects of software. The main objective to engineer the product is to achieve the client’s satisfaction and maintain the quality of software..it specifies the good interaction between client and developer in order to gather requirements. It matters the market trends apart from client and developer communication .Software engineering .It is the establishment and engineering protocols in order to acquire economically obtain software that is reliable and efficiently works in real environment.

1.1 Software Development life Cycle

SDLC is the biography of development of project in order to achieve the objectives and client’s satisfaction .SDLC is the conceptual model which gallant the requirements of client in appropriate model or framework .It consists of process areas each have different protocols and contribution towards development of software. It is s series of steps for modeling the project especially in top-down fashion. In fact, software development life cycle is an systematic approach which deals with every prospects of product in order to make finished product. It is the development model which is the base of software engineering and it maintains the quality, purpose, standards according to intended user. In simply, it is a platform to gallant the ideas of client into appropriate framework. There are various software development life cycle model some of them are:

- Waterfall model
- Prototype model
- Iterative model
- Spiral model

The types of tasks and activities performed in SDLC models are similar , in fact the difference represents the way to engineer the product.



The various activities are:

1. Feasibility –feasibility checks or verifies the preliminary investigation on the type of project. several questions for inquire the project are: is it worth for making the project on level basis like risk exposure, schedule overruns, and client’s satisfaction level.
- 2.Requirements:- The tangible desires for developing the software which are confirmed known at the moment of development partly according to the software requirement specification(SRS).
- 3.Design:- In facts ,the way to plan the needs of predestined client’s and their requirements as finalized according to SRS in pictorial and graphical manner.
- 4.Detailed design:-It includes the logical frame of design with reasonable equitant. The information flows and data flows in between the modules for covering the requirements of product.
- 5.Coding:-To implement the details of design in inquire the questions like why, where, what, etc of modules placed in design for maintaining the objectives of system and to frame into the framework of programming language. The most important is that the developer must be expertise in the specific domain for converting the gal ante of their whole team efforts in order to reach the objectives of proposed system.
- 6.Testing: Testing is the ability for minimizing the risk analysis factor of project .testing should be performed on basis of module and integration of blocks .testing is performed to maintain the reactions of proposed system in exceptions.
- 7.Implementation:-It observes the facility to run the software at user’s environment and developers environment .it checks the final outcomes and maintain the approximation between the actual and observed the outcome .
- 8.Maintainence:-It serves as the satisfaction for the client at their site for using the project and provide mechanism of feedback with the potential feeling that solutions and alternatives are always present in developers mind for evolving the benefits. It maintains the standard protocols as it can’t be lost while re-engineering the product.

2. SDLC Models

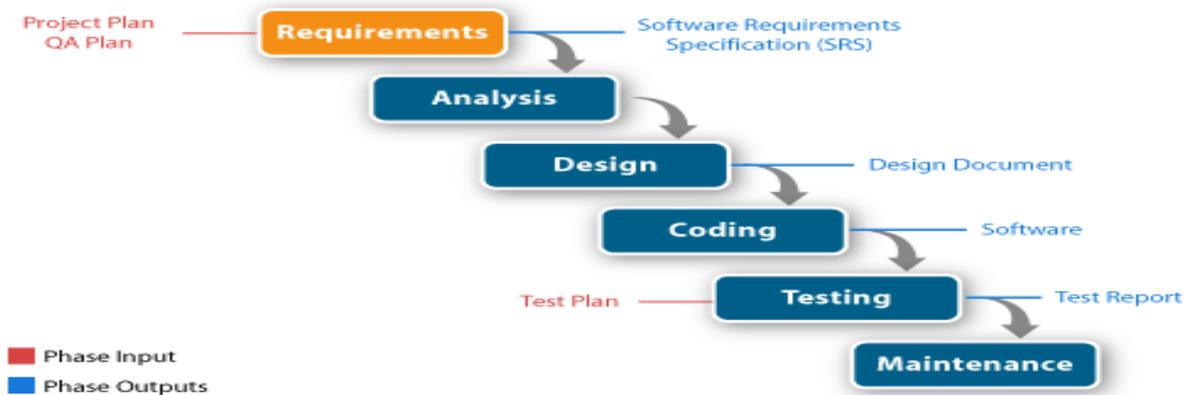
2.1 WATERFALL MODEL

This model is developed by Winston in 1989.in this model. All the activities or whole work is performed in linear fashion “heterogeneous work in homogeneous trend”. Each phase is dependent on one another as the output of one phase becomes the input of another phase and vice-versa. The project plan and quality assurance plan is built prior to requirement phase as it is to check the feasibility of project. The

requirement gathered from internal and external sources are finalized by software requirement specification (SRS). Manual documents are prepared for each phase which helps in dealing with certainty in case of intended client and developer. After executing the design into coding, different test cases are designed to test the modules of proposed project and then test report is generated which decides whether to deliver the project or provide further development. If the project is successfully tested to every aspect then it delivers to client for operating at user's site and provide feedback then only maintenance phase is executed to find alternatives and solutions of feedbacks provided by client. Two important tasks are performed through the whole process development life cycle are:

Verification: the word signifies 'verify' means to check or confirmed the work at each and every linear order is up-to-date acc. To schedule and up-to the mark of achieving the objective intentionally. It executes in every phase.

Validation: -the constitute output of every phases is done acc. to tangible requirements. It simulate the output the " are we build the product right?" in passive mode and patiently check the waged of project process. It executes at the end of software development life cycle.



Merits

1. Resource control is efficient.
2. Manual documentation as maintained of every module in every phase.
3. It is simple and easy to implement.
4. Requirements are well cleared before entering into the next phase.

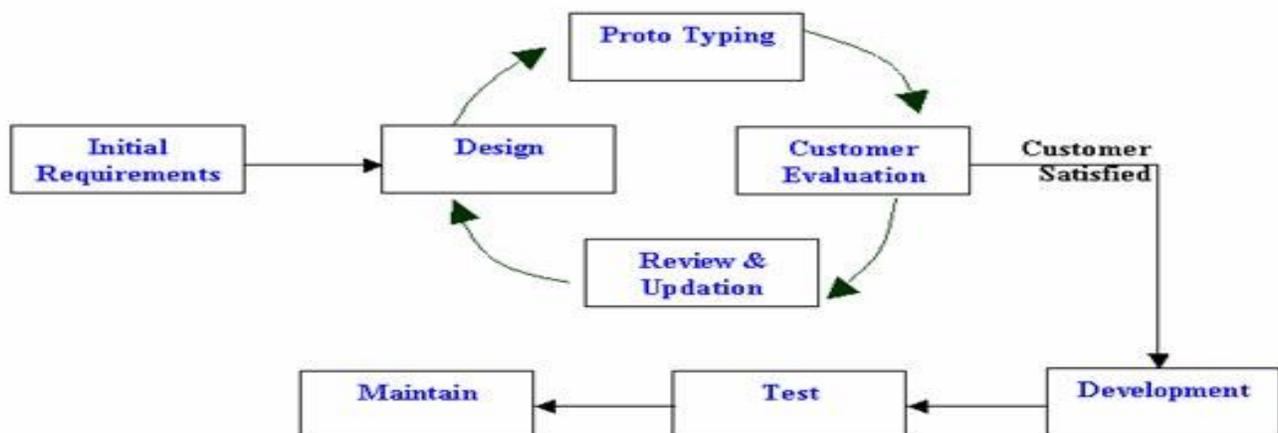
Demerits

1. High levels of risk and uncertainty.

2. Users are not involved at any cost.
3. Users are not fertile with the quality of software and do not aware with performance status.

2.2 PROTOTYPING MODEL

In this model, process emphasizing the quality of waterfall model and prepare a paradigm of eventually known requirements in order to reach “win conditions” of project partly and delivers it to user for providing feedback mechanism. Prototype is an early sample, model or release of a product build to test a concept or process. The requirements are initially known and then prototype is build on the basis of currently known requirement as mentioned in design phase and then it is provided to the customer to take real feel of the project and this process goes on until the client’s is satisfied . If the prototype built is satisfactory acc. The need of plighted user then it transforms into “evolutionary prototype” otherwise “throw-away prototype” in which processed prototype is thrown and again whole process of SDLC is executed for new version of proposed system.



Merits

1. It involves user to extent level **Proto Type Model**
2. Template is built on the basis of steady requirements.
3. User can judge the capabilities of developer.
4. It invites the investors for expanded financial inclusion of project on market basis.

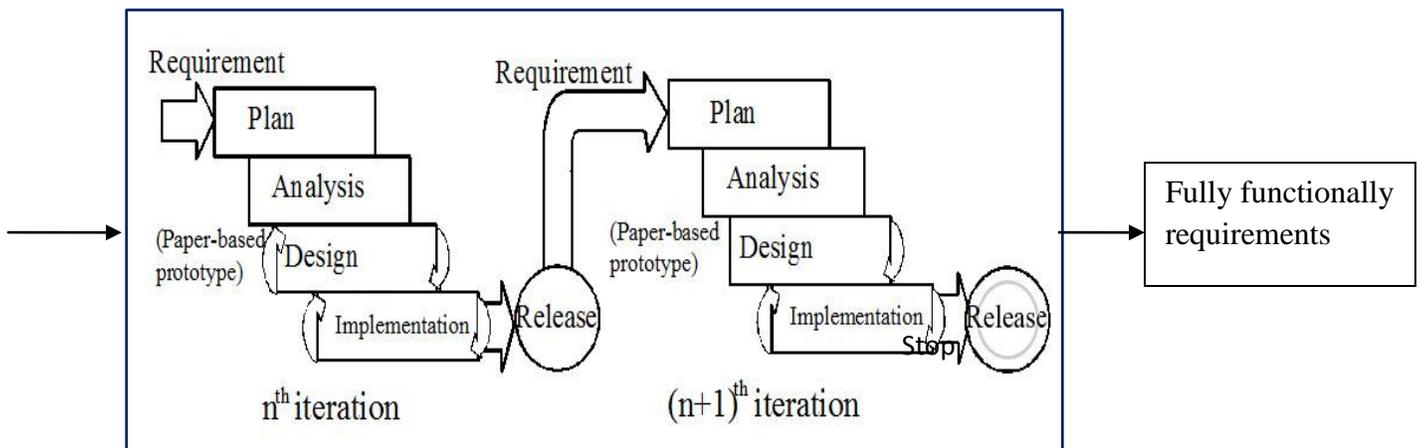
Demerits

1. Extra user involvement may leads to user interruptions in life cycle.
2. Requirements are changed frequently.
3. RAD tools are highly expensive for developers team.

4. Possibility of throw-away prototype.
5. Time consuming and overruns the budget.

2.3 ITERATIVE MODEL

The iterative model is requirement driven software development lifecycle. In iteration model, entire process of SDLC emphasizes on periodic basis. The simple implementation of requirements of proposed system is prepared and delivers to the user for feedback. At each step, requirements are added functionality a each iteration. In other words we can say it processed the requirements at maximum level in the form of iterations or periodical repetitions process. In this iterative model ,if the initial iteration on the basis of initial requirements starts on n^{th} level and it increments on $(n+1)^{\text{th}}$ iterations and this process goes until we get the fully functionally retirements for the proposed project's



Merits

1. Requirement generation phase on functional basis.
2. Failures of module is minimum as testing is appropriately performed on every iteration.
3. Results are obtained early and periodically.
4. Progress status can be easily measured.
5. Minimizes the cost in statistical requirements.
6. Issues , Challenges and risks identified at each increment is utilized and applied to next .
7. It evaluates extra client's evaluation and feedback.

Demerits

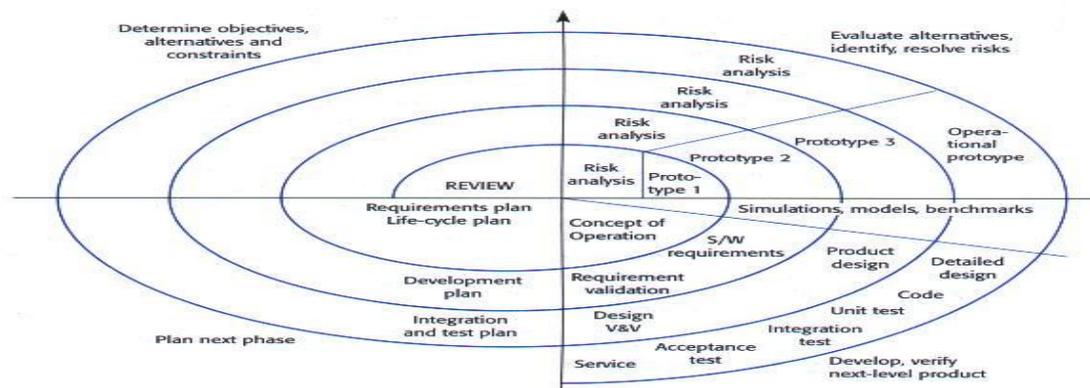
1. It is very consuming process.

2. Management attention is required.
3. Finished end of project may not be judged which can be risk.
4. Integration of iterations diverges from the predefined objective.

2.4 SPIRAL MODEL

Spiral model is risk-driven software development model. It encapsulates the analysis of risk factor includes at every phase of SDLC. It has greater impact on feasibility of project where risk analysis factor is high. It can be a decision support system whether to continue the SDLC process or aborted the project on risk-exposure basis. It starts with first quadrant in order to determining the objectives and constraints imposed on proposed system and then evaluating the alternatives ,identifying the objectives while resolving risks whenever risk analysis is verified by developer then it pursued to next quadrant. The model represents the process of proposed model and simulation represents the performance of model while executing operations over time. Benchmarks is the technique of comparing the benefits of prototypes in order to validate the requirement and design. It performs risk analysis at every aspect of system as testing does not leads to failure of system due to risk.

It works on different milestones.



1. Consider the success conditions of project
2. Identifies the alternatives for finding the solutions and builds ladder to reach the success destination.
3. Evaluate the risk failures at every phase
4. Performs initial operations in order to calculate the cost between constructions and transition of system.
5. Obtain approval for risk evaluation from developers or client's and then pursue to next cycle.

Merits

for developing the project. It also referred as preliminary investigation phase. It deals with different questions:

- Is it worth for developing the project?
- Is it matched with existing project or a new proposed system in particular domain?
- can the budget overruns?
- What is the probability that the project completes on time?
- Can the user involvement converts into interruption?
- What is the type of projects?
- Can the standard for the quality of product is equally maintainable to end level?
- Will be the final product is according to the intended user requirements?

These are several queries that the developer have to process before starting the actual development of project. Feasibility studies is it worth for making the project economically. It imagines when the project will be ready for implementation it will have importance in market. It will stand up to the level of market trends. It signifies the objective of proposed system matched with any predestined existing system so that we can reuse the tools rather than developing again or a newly fresh proposed system which will becomes the challenge for the developer to achieve the objective and deals with client's satisfaction properly. It verifies the observed budget calculated in initial phases is equivalent to the actual budget of finished product. If it will overruns, there is any alternatives to control it as developer commands its team strictly that budget should not be overruns without compromising with the quality of project. It verifies that the developer and his team efforts will deliver the project on time because it the development go out of schedule then it will be the possibility the client denies for accepting the project. There are different types of projects which specifies their risk and failure level like recreational projects, critical , organic ,business or semi-organic projects. The flash points deals with different type of projects are:

- Cost and size estimation(COCOMO Model)
- Explore the market trends
- Explore the technical goals
- Chances of frequently advancements.

These milestones are properly executed so that risk can be easily identified with the help of project monitoring techniques (PMT). This studies continues until the developer is itself satisfied with queries

defined above .There should be an “evaluation team” whose members consists of not more than three persons which evaluates the user’s opinion which clearly defines the artifactory goals and objectives of the project.

Feedback mechanism: Feedback is the facility provided to the client after delivering the product which signifies what is to change, when and why. In this mechanism developer is also included to answer how it will be changed. Sometimes the situation arises when user is continuously giving feedback in a gap of short interval then it becomes inconvenient for the developer .in order to save him ,we derives the formula as if there are n phases for developing the project then only

User’s involvement= $n(n-1)$ feedbacks

If feedbacks are observed more than this then to stopped the process for some time and tries to remind user the rules which have mentioned before starting the project or convince it for successful development of project

Client’s involvement: It signifies the need of “vindicate team” whose members specifies protocols for better communication between client and developer plus for the user to have an eye on the performance of project. Tasks are:

- Objectives are finalized of each step
- It verifies the output of each phase in order to peer the objective before it becomes the input of next phase.
- It has the right to place comment on outcome of phases.
- It verifies the observed outcome of life cycle will appropriately integrated the requirements.
- Testing becomes very easy if efforts of this team is successfully executed as testing is the most difficult and expensive tasks of software development life cycle as half of the cost incurred on this phase

3. SIMULATION OF SDLC-14

Execution of development of software development life cycle-14 implemented in linear ordering of phases. Initially starts with one-pass planning as first one is requirement after dealing with feasibility criteria. In requirements, developers isolates their team members for gathering information of proposed project at user’s environment and uses knowledge accusation tools while hires the expertise members in system domain to share information with each other in order to maximum functionality to requirements. other techniques are also utilized such as interviews the persons working in system domain, questionnaires and objective tasks can also be inducted in order to have answers of probable queries

.Then built a standard document for software requirement specification (SRS).then build a prototype on requirement basis sanctioned be SRS document and makes maximum utilization of vindication team for feedback mechanisms until the client's satisfactory level is achieved. But make sure risk analysis is performed at every prototype and requirements and keep on benchmarking the prototypes to have a evolutionary prototype rather than throw-away prototype. Whenever the prototype is finalized, it is being coded into second –phase of execution which requires strong technical and logical skills of developer. The finalized prototype with customer's satisfaction coded into programming language framework which is economically famous in market trends .then coding of the proposed software is implemented in developer's environment for recording inspection and reviews of issues raised at developer's site. Then that reviews are considered for evaluation which is somewhat nearest to "win conditions". After this testing of particular module by unit testing , integration of module by integration plans and tests cases are performed beta testing is performed at user's site where developer is also present to validate the proposed system .external plus inner details of modules are tested eventually. Now there is the time for implementing the project at user's site in real environment. Maintenance is performed on the feedbacks taken from user's, It has to be decided whether the feedback is corrected or prevented to fall in this situation again or to ignore it. Maintenance also includes reengineer the project when there is a feedback which can't be resolved or to improve the quality of project. Maintenance phase is very expensive and incurred a large share in costs estimation of project. Documentation of every phase is done in standard language which is purely understandable by experts. Risk analysis is performed in every context .The teams like "evaluation team and "vindication team" performs their tasks efficiently and accurately. A Software will definitely completed on time with best quality and happy client's.

Tabular form of merits and demerits of SDLC_14

MERITS OF SDLC-14	DEMERITS
Requirements are well known at initial phases	It is somewhat not reliable for very large projects
It is very easy to implement and eco-friendly	It is little bit expensive
It guarantees about95% to achieve the success	
User involvement is at higher level	
Less chances of software's crisis	
Resource control is efficient	
Client is approximately satisfied	

It maintains the quality of project	
It is flexible and reliable	
Time frame is also short	
It maintains the budget and delivers the project on schedule	
Access control is maximum.	
Teams are developed for efficient functioning	

4. CONCLUSION

The development lifecycle is done by sdlc-14 which counters the maximum profit rather than implementing several software development lifecycle models. Small and medium large projects are easily developed with SDLC-14. It achieves the client satisfaction and quality of project at extreme level free of risks.

5. REFERENCES

1. K.K Aggarwal , Yogesh Singh Software Engineering 3rd Edition.
2. Software Development Life Cycle(SDLC)- the five common principles.htm.
3. Roger Pressman titled Software Engineering –a practitioner’s approach.
4. Analysis of Evolution of New Software Development Life Cycle Model SDLC-2013 with Client satisfaction. Volume-3 Issue-1 by Naresh kumar and Abhinav Shukla
5. www.en.wikipedia.org/wiki/system_development_life_cycle.
6. K.Schewalbe(2009),”Information Technology project Management”,6th Edition, Cengage Learning.
7. ALAN M Davis, Senior Member, IEEE, EDWARD H. BERSOFF, Senior Member, IEEE and EDWARD R. Corner, Member, IEEE” A Strategy for comparing alternatives software development life cycle models ”IEEE Transaction on Software Engineering,vol.14,no.10,October 1988.