A NEW SDLC FRAME WORK FOR WEBSITE DEVELOPMENT

Simon Kevin John[1]
Swapna.K (Kannankott)[2]

Abstract

SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality software. There are various software development life cycle models defined and designed which are followed during 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, in order to ensure success in process of software development. In today’s business having a web site or customer suite is very important for smooth functioning of its operations and due to the aggressive competition in the market .It is significant to select a SDLC model that will not only allow the team to meet the client requirements but also provide high quality software (website) in the least amount of time. This paper focuses on developing a new SDLC frame work to suit the dynamic and highly automated environment of today

Keywords: SDLC, New Fame work, Web site development

Introduction

Software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. 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. Any software development process is divided in several logical stages that allow a software company to organize its work efficiently in order to build a software product of the required functionality within a specific time frame and budget. Every day, we use the internet and visit different websites to learn, to shop, to do business or even have fun. Website creation also takes planning and SDLC could also be applied to Web Development. All software projects go though the phases of requirements gathering, business analysis, system design, implementation, and

[1] Student (MMS) ,YMT College of Management, Kharghar, Navimumbai
[2] Assistant Professor , YMT College of Management, Kharghar, Navimumbai

quality assurance testing. Employing any SDLC model is often a matter of personal choice entirely dependent on the developer. Each SDLC model has its strength and weakness, and each SDLC may provide better functionalities in one situation than in another. Then the challenge is to decide which model should be selected o provide a particular set of functionalities under certain circumstances. To be successful in today's dynamic business environment, it is very important to adopt a best possible method by which the time required for development can be reduced and at the same time product quality and customer satisfaction can be improved.

Today's fast-paced competitive business environment is characterized by automation and innovation. Now, more than ever, the ability to challenge the norm (be it process models) is of vital importance. Examining alternatives from a range of viewpoints in order to critically appraise the situation is essential to business success.

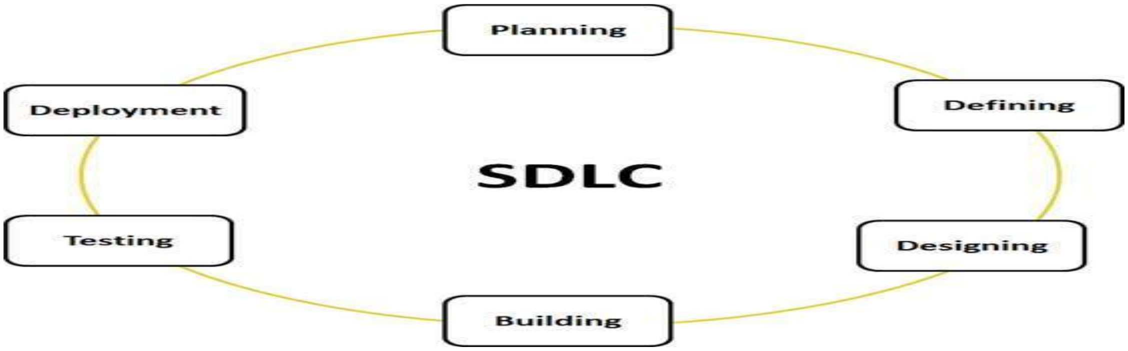
Most web application development relies on the experience of individual developers without any rigorous or systematic approach (Gellersen, 1999). There is a high demand for a systematic approach to developing and managing web information systems. There are many different methodologies being used to suit the needs of programmers but no one standard methodology that is accepted by all the software engineering community (Escalona et al, 2002). This research paper focusing on developing a new SDLC frame work to reduce the time taken for development while maintaining quality and productivity .This proposed model is adapted from existing methodologies and applied to the context of web development. This paper will outline in detail the proposed phases of the new SDLC framework.

Phases involved in SDLC Model

The phases that are generally present in each and very software development life cycle model are

1) Planning and Requirement Analysis
2) Defining Requirement
3) Designing the product architecture
4) Building or developing the product
5) Testing the product
6) Deployment in the market and maintenance

The following figure is a graphical representation of the various stages of a typical SDLC.



Explanation of the phases

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through .SRS. . Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the product architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity , budget and time constraints , the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java, and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometime product deployment happens in stages as per the organizations. business strategy. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during 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, in order to ensure success in process of software development.

Objectives of this Research

- ▶ To identify the bottlenecks in project (Website) development life cycle.
- ▶ To Study the methods to minimize the impacts of bottlenecks on Project (Software) development life cycle.
- ▶ To conduct a comparative analysis of different life cycle models in use for project development.
- ▶ To understand the competitive dimensions for the company and to provide competitive edge over the competitors.

Literature Review

- In the software Industry the hybrid of different methodologies (waterfall, RAD, Spiral) is used i. e with some modification. In this paper, we have compared the different software development life cycles models on the basis of certain features like requirement specification, risk involvement, use involvement, cost etc. On the basis of these features for a particular project one can decide which of these software development life cycle models should be chosen for that particular project .Selecting the correct life cycle model is extremely important in a software industry as the software has to be delivered within the time deadline and should also have the desired quality (A comparative study of different software development life cycle models in different scenarios by Mr.Apoorva Mishra and Deepty Dubey –International journal of advance research in computer science an management studies (Volume 1,issue 5 ,October2013)

- In the present scenario all software systems are imperfect because they cannot be built with mathematical or physical certainty, Hence in this research paper the comparison of various software development mode s has been carried out. According SDLC each and every model have the advantage and drawbacks so in this research we have to calculate the performance of each model on behalf of some important features. The concept of system lifecycle models came into existence that emphasized on the need to follow some structured approach to wards building new or improved system. Many models were suggested like waterfall, Iterative model, prototype model, spiral model etc. After analysis of all models through the various factors ,it has been found that the original water fa ll model is used by various big companies for their internal projects .Since the development team is familiar to the environment and it is feasible to specify all requirements of working environment. Iterative water fall model overcome the drawback of original waterfall model .It allow feedback to proceeding stage. Prototype model used to develop online systems for transaction processing. Since significantly reduce rework and lead to the creation of working model in lower capital cost. Spiral model is used for development of large, complicated and expensive projects like scientific Projects .Since spiral model approach enables the project team to address the highest risk at the lowest total cost (A Comparative Analysis of Different types of Models in Software development Life Cycle by Ms. Shikha maheshwari and Prof.Dines h Ch. Jain, International Journal of Advanced Research in Computer Science and Software Engineering)

- A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. But none of the SDLC models discuss the key issues like Change management, Incident management and Release management processes within the SDLC process, but, it is addressed in the overall project management. In the proposed hypothetical model, the concept of user -developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. In the proposed hypothetical model, the concept of user -developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. The ―one size fits all‖ approach to applying SDLC methodologies is no longer appropriate. We have made an attempt to address the above mentioned defects by using a new hypothetical mode l for SDLC described elsewhere. The drawback of addressing these management processes under the overall project management is missing of key technical issues pertaining to software development process that is, these issues are talked in the project management at the surface level but not at the ground level (Evolving A New Model (SDLC Model-2010)For Software Development Life Cycle (SDLC)PK Ragunath,S.Velmourougan ,P. Davachelvan S.Kayalvizhi,R.Ravimohan , IJCSNS International Journal of Computer Science and Network Security , VOL.10 No.1, January2010)

- In recent years, software testing is becoming more popular and important in the software development industry. Indeed, software testing is a broad term encircling a variety of activities along the development cycle and beyond, aimed at different goals. Hence, software testing research faces a collection of challenges. A consistent roadmap of most relevant challenges is proposed here. In it, the starting point is constituted by some important past achievements, while the destination consists of two major identified goals to which research ultimately leads, but which remains as reachable as goals. The routes from the achievements to the goals are paved by outstanding research challenges, which are discussed in the paper along with the ongoing work. Software testing is as old as the hills in the history of digital computers. The testing of software is an important means of assessing the software to determine its quality. Since testing typically consumes 40~50% of development efforts, and consumes more effort for systems that require higher levels of reliability, it is a significant part of the software engineering .Software testing is a very broad area, which involves many other technical and non -technical areas, such as specification, design and implementation, maintenance, process and management issues in software engineering. Our study focuses on the state of the art in testing techniques, as well as the latest techniques which representing the future direction of this area. Today, testing is the most challenging and dominating activity used by industry, therefore, improvement in its effectiveness, both with respect to the time and resources, is taken as a major factor by many researchers The purpose of testing can be quality assurance, verification, and validation or reliability estimation. It is a tradeoff between budget, time and quality. Software Quality is the central concern of software engineering. Testing is the single most widely used approach to ensuring software quality. (A Research Study on importance of Testing and quality Assurance in Software Development Life Cycle (SDLC) Models by Maneela Tuteja , Gaurav Dubey International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3,July 2012)

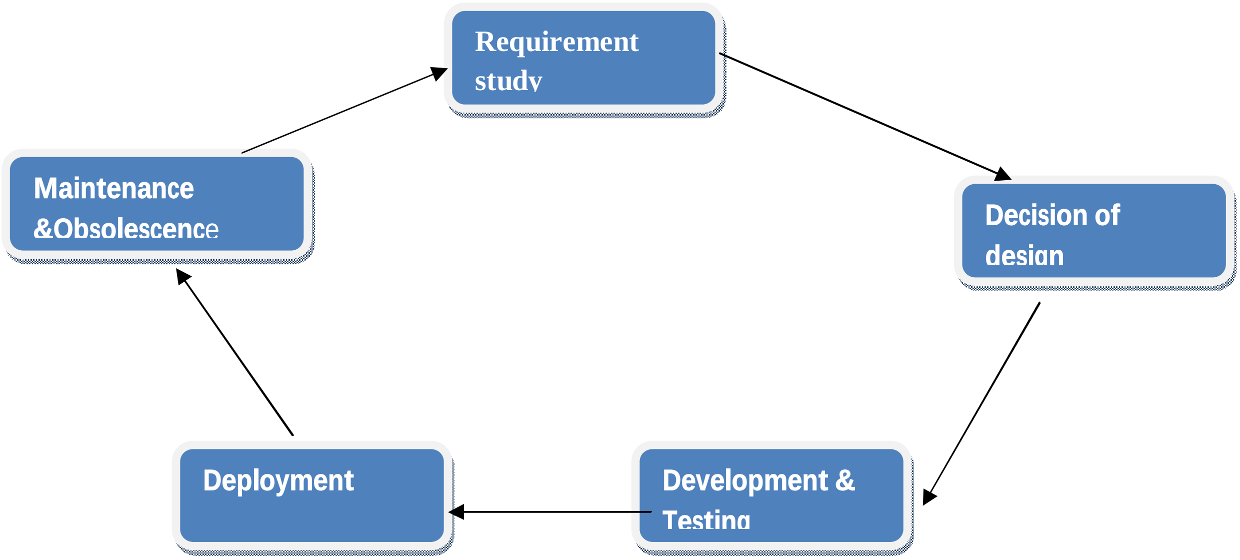
Analysis

- It can be understood that most of the projects take less than 2 months to complete.
- Requirements gathering is considered to be the most important phase followed by the testing
- The most time taking phase is coding
- Alternatives are highly considered before development begins
- Change management is an important issue
- Risks were identified at each and every phase of the SDLC
- Communication was said to be one of the most crucial point for success.

Suggestions

- The phases should be:-
    - Requirements Study
    - Decision of Design
    - Development & Testing(Simultaneously)
    - Deployment
    - Maintenance & Obsolescence

Figure: New SDLC Framework for Website Development



Phase 1: Requirements Study

The definition of the product must be very clear and more time must be given to this phase and when this phase is over, it is important to verify whether the following questions have been answered:

1) What features are important?
2) What do the users need it to do?
3) What restrictions (site, environment, material availability) are there?
4) What software do we need?
5) What hardware do we need to run it?
6) What connectivity do we need?

Phase 2: Decision of design

This phase mainly focus on deciding which existing design can be used for fulfilling the customer requirements for proposed website .since today's environment is highly automated ,existing designs can be effectively utilized for even for customization. Questions that should be answered for this phase are

1) What do we need the system to output?

2) What inputs do we have available?
3) Which existing design can be leveraged for this?

Phase 3:Development and Testing

Coding should be highly prioritized and more time should be given for his phase and along with coding the testing should also be simultaneously completed therefore more time should be assigned for this phase and the questions that should be answered for this phase are :

1) Put all the objects gathered in the design phase?
2) Whether redesign possible if the system does not work?
3) Do we have sufficient training materials and documentation?

Phase 4 :Deployment

During deployment complete testing and training of the staff should be done and reviews must be taken from the client periodically. Important questions for this phase are:

1) Are the users and support staff trained?
2) Hat bugs are there?
3) Does everyone agree that the system meets the requirements?

Phase 5: Maintenance and Obsolescence

Maintenance

Periodic maintenance is a must and these questions must be answered during maintenance:

1) What new features are requested?
2) What new regulations dos it need to comply to?
3) What additional training is required?

Obsolescence

Precautions should be taken in case of the obsolescence of the software and all the safety precautions for obsolescence of the software should be taken.

Limitations

The above model will be affective for smaller startup industries

Conclusion

For the above suggested model to be successful it is very important that there must be good communication between the developer and the project manager and the client

References

1. Dr. Deepshikha Jamwal "Analysis of Software Development models"
2. Roger S pressman " Software Engineering – A practitioner's Approach"
3. A Research Study on importance of Testing and quality Assurance in Software Development Life Cycle (SDLC) Models by Maneela Tuteja , Gaurav Dubey International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3,July 2012
4. Evolving A New Model (SDLC Model-2010)For Software Development Life Cycle (SDLC)PK Ragunath,S.Velmourougan ,P. Davachelvan S.Kayalvizhi,R.Ravimohan , IJCSNS International Journal of Computer Science and Network Security , VOL.10 No.1, January2010
5. A comparative study of different software development life cycle models in different scenarios by Mr.Apoorva Mishra and Deepty Dubey –International journal of advance research in computer science an management studies (Volume 1,issue 5 ,October2013