|  |  |  |
| --- | --- | --- |
| **COURSE NAME: SOFTWARE ENGINEERING**  **COURSE CODE:**  **SUBMITTED TO:**  **Mr. RAVI SHARMA**  **SUBMITTED BY:**  **ABHINAV KUMAR CHOUDHARY(21SCSE1011615)**  **ABHIJEET KUMAR MISHRA(21SCSE1010930)** | | |
| **“A Case of Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model”**  **“ABSTRACT”**  Software Development Life Cycle (SDLC) Models are the frameworks used to design, develop and test the software project. The models are set of procedures which are to be followed during the software development process. These models make sure that the software development is according to the needs of the customer and insure software will design within the given timeframe and budget. There are many SDLC models used during software development process. These models are also referred as Software Development Process Models (SDPM). Each process model follows a series of steps, to ensure success in the process of software development. We have different types of SDLC models. The SDLC models are waterfall model, iterative model, spiral model, V-model, agile model, RAD model and prototype model. Each of these models has its own weak point and strengths. In this paper a new model proposed called Z-SDLC model for software development that lays special emphasis on client satisfaction and also tries to fulfil the objective of the Software Engineering for the development of high quality software product within timeframe and budget. The new proposed model is designed in such a way that it allows client and software company to interact freely with each other in order to understand and implement requirements in a safer way.  **INTRODUCTION**  Software Development Life Cycle (SDLC) is a procedure by which quality software project can be developed within timeframe and budget and also according to the client’s expectations and prospects. SDLC ensures quality of software project. All software development processes models include various activities like requirements gathering, system feasibility, system analysis, system design, coding, testing, implementation and maintenance. The software company or the team of software developers have choice to select the SDLC model. Each of these models has its own weaknesses and strengths in different situations and circumstances. The challenge is to select which model should be good under certain conditions. Most of the present SDLC models have little attention towards client satisfaction. It matters not only to the client but even more to the software company because it costs far less to retain a happy client than it does to find a new client. Satisfying client is main for staying in the market and in modern world of global competition. Client satisfaction is very necessary for the acceptance and delivery of the software project. Software projects fails in the absence of client satisfaction. SDLC model must fulfill the requirements of the client/customer as per there expectations and even delight with the value of quality services.  **SOFTWARE DEVELOPMENT LIFE CYCLE**  Software Development Life Cycle (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 change or enhance specific software. The SDLC aims to produce a high quality software that meets or exceeds client expectations, reaches completion within times and cost estimatesThe 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 following figure is a graphical representation of the various stages of a typical SDLC.   * Planning and Requirement Analysis * Defining Requirements * Designing the Product Architecture * Building or Developing the Product * Testing * Deployment * Maintenance  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  **B. 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 Software Requirement Specification (SRS). SRS document which consists of all the product requirements to be designed and developed during the project life cycle   1. **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 Design Document Specification (DDS). 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 .  **D. 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. 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.    **E. 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 .  **F. Deployment**  In the Market 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 User acceptance testing (UAT).  **G. Maintenance**  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 (SDPM). Each process model follows a series of steps unique to its type, in order to ensure success in process of software development.  Following are the most important and popular SDLC models followed in the industry:  A. Waterfall Model  B. Prototype Model  **“Waterfall Model”**  The Waterfall Model was first process model to be introduced provided by Winston W. Royce in 1970. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In the waterfall approach, the whole process of software development is divided into separate phases. In waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.  Following is a diagrammatic representation of different phases of waterfall model.  1. Requirement Analysis  2. System Analysis  3. Implementation  4. Verification  5. Maintenance | | |
| Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:  **♣ Requirements are very well documented, clear and fixed .**  **♣ Product definition is stable.**  **♣ Technology is understood and is not dynamic.**  **♣ There are no ambiguous requirements .**  **♣ Ample resources with required expertise are available to support the product.**  **Prototype Model**  The Software Prototyping refers to building software application prototypes which display the functionality of the product under development but may not actually hold the exact logic of the original software. Software prototyping is becoming very popular as a software development model, as it enables to understand customer requirements at an early stage of development. It helps get valuable feedback from the customer and helps software designers and developers understand about what exactly is expected from the product under development. Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation. Prototyping is used to allow the users evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.  Following is the stepwise approach to design a software prototype:  ♣ Basic Requirement Identification  ♣ Developing the initial Prototype  ♣ Review of the Prototype  ♣ Revise and enhance the Prototype  Software Prototyping is most useful in development of systems having high level of user interactions such as online systems. Systems which need users to fill out forms or go through various screens before data is processed can use prototyping very effectively to give the exact look and feel even before the actual software is developed. Software that involves too much of data processing and most of the functionality is internal with very little user interface does not usually benefit from prototyping. Prototype development could be an extra overhead in such projects and may need lot of extra efforts.  **“NEW PROPOSED Z-SDLC MODEL”**  The new Z-SDLC model is planned in such a way that it allows software company and client to freely interact with each other in order to understand the requirements of software project in a good way to develop a good quality software within a given timeframe and budget. SDLC process model start with the client’s requirements so the proposed model tries to find every requirements like functional requirements, non-functional requirements and user requirements of the client/customer. It helps in developing a good quality of software product that satisfies the client/customer needs. The scope of computer based system products, client satisfaction is very much dependent on how system development process works to build operational product that satisfy the client’s need and also related with the expected requirements. Finally, client satisfaction depends upon the good understanding about the client needs and associated user requirements for a better software product and the capability to connect those requirements to the software company  My proposed Z-SDLC model include the following:   * **SYSTEM ANALYST TEAM (Requirement Gathering Plan)**   The System Analyst team have a sufficient knowledge of computer science, software engineering, software development processes, software applications, operating system, as well as domain knowledge like various business functions to be performed. The system analyst team coordinates with the risk factor team and technical team. System Analyst team deals with the client for Identify Problem, Breakdown Requirements, Make a Prototype, Finalize the Requirements, Feasibility Study, Approval of SRS and any ambiguity of client is also discuss and solved by the system analyst team.   * + Identify the Problem   + Identify the Requirements   + Breakdown Requirements   + Finalize the Requirements   + Feasibility Study   + Approval of SRS Document   + Make a Prototype * **RISK FACTOR TEAM (Success Approval Plan)**   What is Risk?  Risk are future uncertain events with a probability of occurrence and a potential for loss. Risk identification and management are the main concerns in every software project. Effective analysis of software risks will help to effective planning and assignments of work . Risks are identified, classified and managed before actual execution of program. These risks are classified in different  categories.   * Schedule Risk * Budget Risk * Operational Risks * Technical risks * Programmatic Risks   The risk management in SDLC model is started as per the client’s requirements. In the beginning, these requirements are in the mind of the client. The system analyst team by using a software development model has to identify, discover, understand and fulfill the requirements of the client in order to satisfy the client. The requirement phase of the Software Development Life Cycle transforms the idea in the mind of the client into a formal document known as Software Requirement Specification (SRS). The quality of the SRS impacts client satisfaction, system validation, quality of final software, software development cost and schedule.   * TECHNICAL TEAM (Development Plan)   Technical team is an expert team and its team members are updated with new technologies and new software products. It is a technically expert team. This team interacts with system analyst team during its working. Technical team studies the SRS document (the requirements document) received from the system analyst team which in turn get these requirements from the client/customer. The member of technical team is full of skills and interacts with system analyst team. Technical team works on non-developed requirements. After feasibility study and risk analysis the technical team finally verify the final SRS document, check the prototype provided to the client/customer and start work on the following phases, i.e. Designing, Coding, Testing, Implementation, Maintenance each of these phase also followed by validation process.  1. Designing  2. Coding  3. Testing  4. Implementation  5. Feedback and Maintenance   * **Software Development by Waterfall model**   The waterfall model is a linear sequential model. We considered the requirements, check them and moved towards the designing phase followed by the coding and testing phases Technologist Lab In-charge (the client) because they were not satisfied. As the client want to change it in terms of graphics, functionality and features the waterfall model does not allow to change after completing the requirements so, it fails to convey client about the software product.   * **Software Development by Prototype model**   The prototype model build prototype to give feel of the proposed software to the client. As we already have Logical LIMS requirement so, we build prototype and showed it to the client. After client’s feedback, we changed it and again showed it to the client. After building and showing three prototypes, client finalized the requirements and we passed these final requirements to next phases for software development and named it as LIMS-2. Finally LIMS-2 was delivered to the client. But building prototype affects cost, schedule and effort which get exceeded.   * **Software Development by Z-SDLC model**   Z-SDLC model is a new model for the software development. The striking feature of this model is the client satisfaction. Firstly, system analyst team deal Technologist Lab In-charge (the client) to discover the problem and requirements. After discovering the problem the system analyst team breakdown this requirement into developed requirements and non-developed requirements. The system analyst team finalize the requirement and also start work on the feasibility study and SRS document. The system analyst team with the collaboration of technical team analyzed the available requirements provided by the system analyst that is present in SRS document for the proposed system and searched the most matching software for them. Now the system analyst team showed the software to the client so that the client got the feel of proposed software and also identifies the undiscovered requirements and gave his suggestion and feedback to the system analyst team. The system analyst team again passed these suggestions to the technical team and the process goes on until the client finalized the requirements. System analyst team passed final requirements to the risk factor team for the risk analysis and the requirement validation. After validation and resolving various risk associated with the final requirements, these final requirements were passed to technical team for final software product. The technical team start work on the following phases i.e. designing, coding, testing, implementation and maintenance followed by the validation process to develop the final product named as Logical LIMS. Logical LIMS was approved by the client because it satisfied the client’s requirements within budget and given timeframe because budget and timeframe were not disturbed or affected due to various increments or by building prototype. Finally, User Acceptance Test is signed with the client.  **CONCLUSIONS**  In this research paper various models like a waterfall and prototype models have been considered and various topographies like requirement specification, cost, risk factor, user involvement, success rate, simplicity is analyzed. Each model has its own pros and cons. In the requirement gathering phase the software developer can select the suitable software development life cycle model according to the needs. My suggested work can be concise as the construction of the new Z-SDLC model for more efficient software development. The goal of the Software Engineers are to grow the software industry at bigger stage and want to make software with high quality within budget and schedule. My suggested plan tries to accomplish the objective of Software Engineers by showing existing matching software as a prototype to the client/customer for discovering the requirements efficiently from the client/customer in order to approximation of cost, time frame, schedule, work and effort more accurately and precisely.  **“REFERENCES”**  [1] Roger S. Pressman, Ph.D. “Software Engineering a Practitioner’s Approach, Fifth Edition: McGraw-Hill Higher Education, 2001, pp. 53-193  [2] Roger S. Pressman, Ph.D. “Software Engineering a Practitioner’s Approach, Fifth Edition: McGraw-Hill Higher Education, 2001, pp. 245-507  [3] Ian Sommerville, "Software Engineering", Addison-Wesley, 2007  [4] The Software Development Life Cycle (SDLC) for Database Applications, First Edition, Digital Publication LLC 2005, pp. 14-21.  [5] Rod Stephens, "Beginning Software Engineering", John Wiley & Sons, 23-Mar-2015  [6] Elvis Foster, "Software Engineering: A Methodical Approach", Apress, 16-Dec-2014  [12] SDLC Overview Tutorials Point Simple Easy Learning. Available: <https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>  [13] SDLC Overview Tutorials Point Simple Easy Learning. Available: <https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>  [14] SDLC Overview Tutorials Point Simple Easy Learning. Available: <https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm>  [15] SDLC Overview Tutorials Point Simple Easy Learning. Available: <https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm>  [16] Types of Risks in Software Projects, Basics of Software testing, Quality assurance, Test strategy | Last Updated: "December 14, 2016". Available: http://www.softwaretestinghelp.com/types-of-risks-in-software-projects/ | | |