

SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

Rahul Yadav, Rajeev Ranjan
*Student, Department of Information Technology
Dronacharya College Of Engineering*

Abstract- In present all software systems are not perfect because they cannot be built with mathematical or physical certainty. In the era of software development there exist an outsized variety of Models to develop software. Each model has its own characteristics, limitations and dealing surroundings. In step with the wants, software industry people use completely different models to develop different software. There are numerous models however none of them is capable to address the problems of client satisfaction. Thus in this research paper the comparison of various software development models has been carried out. According SDLC each and every model has the advantage and drawbacks so in this research we have to calculate the performance of each model on behalf of some important features. Many models were suggested like waterfall, Incremental model, spiral model etc. We also discussed new free flow SDLC model for software development.

Index Terms- Software Development Life cycle (SDLC), Software lifecycle, and software development.

I. INTRODUCTION

A software development method, conjointly referred to as a software development life cycle (SDLC), could be a structure imposed on the development of a software product. It is often thought-about as a set of system development life cycle. There are many models for such processes, each describing approaches to a range of activities that take place throughout the method.

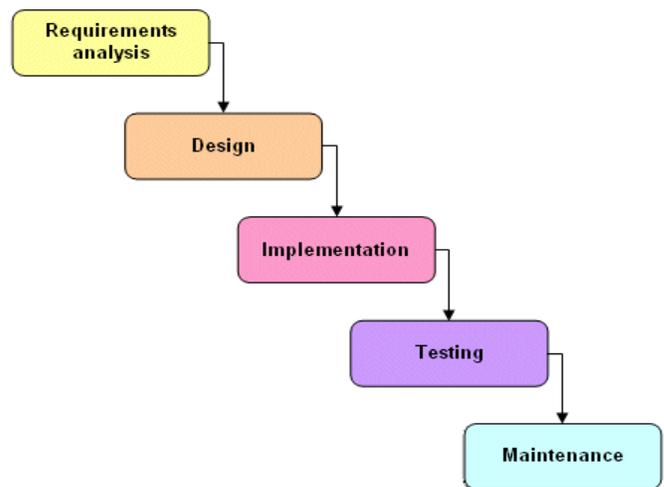
A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. A descriptive model describes the history of how a particular software system was developed. Descriptive models may be used as the basis for understanding and improving software development processes or for building empirically grounded prescriptive models. None of these models deals with the concept of rigorous requirement gathering management and robust testing techniques. Requirement gathering is the process through which

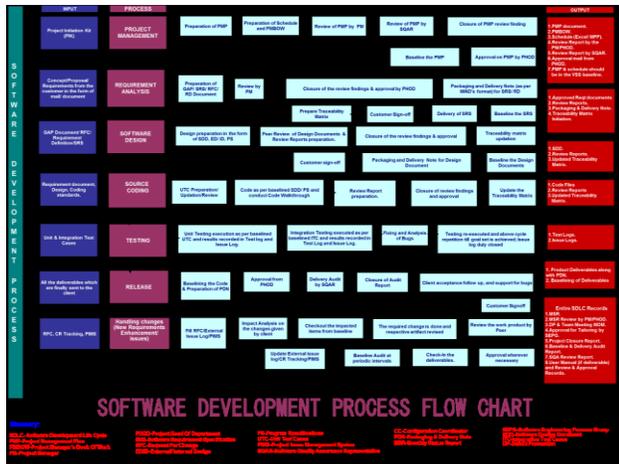
requirements of the software to be built is gathered and obtained by its users and Testing is the technique wherein the developed software is tested through different phases to check the stability and robustness of the software. In this paper, we are dealing with and augmenting different phases of Software Development Lifecycle Model (SDLC) which helps us in providing an efficient, reliable, easy to implement and effective during and after implementation.

II. SOFTWARE DEVELOPMENT MODELS

(1) Waterfall Model - The Waterfall model is a conventional, linear, sequential or traditional waterfall software life cycle model. It is a sequential development approach, in which development is seen as flowing steadily downwards through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

Advantages – It is simple to use and understand and





activities and modify them according to the requirement of the project. **Free-Flow Model** eliminates the risks and errors in early stages through pipelined flow of processes and control. Complete architecture of Free-Flow Model is shown as below in Figure.

FREE-FLOW SDLC FEATURES

Requirement Capture We capture requirements through conference calls, interfacing with on-site resources, client side visits and studying existing code and implementations.

Prototype and High Level Diagrams Documents are always necessary in a well-defined process but it is very difficult for customers to figure out upfront what they will get for their money. Prototype and High level diagrams help the clients to have a clue of what will be delivered after coding.

Module releases - Every project is divided into multiple modules. As soon as a module is completed we send the demonstration version to the customer. This enables customers to track progress and notify us with any change in flow.

Check-List Method From planning to development, drastically reducing the commonly made mistakes by developers while coding, releasing and deploying

Well-Defined Architecture The above architecture helps us to train newly recruited developers and leaders quickly. The responsibilities are clearly defined at every stage which helps in improving the efficiency during the development.

III. PERFORMANCE EVALUATION

FEATURES	WATERFALL	SPIRAL	INCREMENTAL	FREE-FLOW MODEL
UNDERSTANDING REQUIREMENTS	WELL UNDERSTOOD AT BEGINNING	NOT WELL UNDERSTOOD AT BEGINNING	WELL UNDERSTOOD AT BEGINNING	WELL UNDERSTOOD AT BEGINNING
COST	LOW	HIGH	MEDIUM	LOW
SCHEDULE	WITHIN SCHEDULE	SCHEDULE MAY EXCEED	SCHEDULE MAY EXCEED	WITHIN SCHEDULE
RISK INVOLVEMENT	HIGH	LOW	MEDIUM	VERY LOW
USER INVOLVEMENT	LOW	HIGH	HIGH	HIGH
GUARANTY OF SUCCESS	LOW	GOOD	HIGH	HIGH
CLIENT SATISFACTION	LOW	HIGH	HIGH	HIGH
FLEXIBILITY	RIGID	FLEXIBLE	FLEXIBLE	FLEXIBLE
TIME FRAME	MEDIUM	SHORT	VERY LONG	SHORT
INITIAL PRODUCT FEEL	NO	YES	NO	YES

IV. CONCLUSION

After analysis of all models through the various factors, it has been found that the original water fall 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. Spiral model is used for development of large, complicated and expensive projects like scientific Projects .Since spiral model approach enables the project term to address the highest risk at the lowest total cost. The proposed work can be summarized as the creation of the approach FREE-FLOW SDLC to develop software more efficiently. The aim of Software Engineering is to develop software of high quality within budget and schedule. The proposed plan tries to fulfil the objective of Software Engineering by defining a well-defined procedure and targets for the client to discover the requirements efficiently from the client in order to estimate cost, schedule and effort more accurately and map them

further in the development lifecycle. With the proposed work, the effectiveness of optimization has been studied carefully. Further investigation to the topic reveals that FREE-FLOW SDLC can give good results. The concept has been worked out and can be used in future.

REFERENCES

- [1]http://www.idt.mdh.se/kurser/ISD_cdt417/files/articles/paper5.pdf
- [2]http://www.ijarcsse.com/docs/papers/Volume_3/9_September2013/V3I9-0191.pdf
- [3]http://www.ijarcsse.com/docs/papers/May2012/Volume2_issue5/V2I500405.pdf
- [4] <http://software-security.sans.org/resources/paper/cissp/comparison-software-development-lifecycle-methodologies>
- [5]https://www.google.co.in/?gfe_rd=cr&ei=471VVNL6BcnM8genzYCQCw&gws_r=ssl#q=advantage+and+disadvantage+of+incremental+model