iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.38736

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Comparative Study of Different SDLC Models

Aparna Gupta[1], Achint Rawal[2], Yamini Barge[3]
[3]*Assistant Professor,* [1, 2]*Department of Computer Science and Engineering, Acropolis Institute of Technology and Research*

Abstract: ***The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software which are cost effective and reliable. This method of software developing process is quite systematic and structural. SDLC defines the framework that has different activities and tasks to be administered during the software development process.***
Software development process is quite complex, and to do it without any proper planning would be inefficient. So, we use these SDLC models to make the Software development process simple and systematic. There are various software development life cycle models that are used in the software development process, all having their own advantages and limitations. In this paper, we have included six of these SDLC models - Waterfall Model, Spiral Model, V Model, Agile Model, Iterative Model and Rapid Application Development (RAD) Model. These Software Development processes have their own Advantages and Disadvantages, and the main purpose of this paper is to explain these models and know the difference between them.
Keywords: Software Development life cycle, Models, Risk Analysis, Framework, SRS.

I. INTRODUCTION

The software project planning management is the clear need in today's era. An idea is sort of a road map. Software engineering is a discipline for developing top quality software for computer-based systems. Software engineering is all about the standard focus, methods, and tools that are utilized in software development process.
The software development process defines a framework that has different activities and tasks. We will see what different activities and tasks are to be administered during software development.

We have tried to explain the phases of a typical SDLC process in a precise manner, which is give below –

A. Planning

Planning and requirement analysis is the fundamental but the most significant phase of each life cycle process. It is brought to completion by the senior members, after conducting meetings with customers or the clients. Quality assurance requirement, risk identification and feasibility report are the most crucial outcomes of this phase.

B. Defining Requirements

When the process of planning and requirement analysis is completed, then the subsequent phase is the one in which detailed description of the necessities and the documentation of these requirements is done. Further, validation from the client(s) is obtained. The output of this phase is the software requirement specification (SRS) document which contains all the wants of product to be designed and developed.
Design Requirement includes two distinct activities:

1) *Requirements Analysis:* In requirement analysis, all related data and knowledge of customer requirements are collected and analysed.
2) Software Requirement Specification: After analysing requirements of the user, software requirement specification (SRS) is developed. In SRS, the entire road map for the project is defined, which describes the activities to be carried out during the course of the whole project.

C. Designing and Software Architecture

Software requirement specification is employed as an input to style the architecture of the software package which is being developed. Initially, the architecture is designed, which is then reviewed by important stakeholders. The architecture consists of criteria like risk assessment, robustness, design modularity, time duration, cost etc. and then, the simplest and most efficient design is selected.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

D. Building or Developing the Product

In this phase, actual development of the product starts according to the designed architecture and the proposed road map. If designing is completed successfully, then this phase isn't much difficult. Developers use different tools like compilers, interpreters and debuggers to generate code. Different programming languages like C, C++, Pascal, Java and PHP are used, depending on the need. Programming language depends upon type of software being developed.

E. Testing

In this phase, developed product is tested on the basis of whether it meets the user requirements, which are clearly mentioned in software requirement specification (SRS) document. Software defects are reported, tracked, fixed and retested, in order to provide top quality product to the concerned client or organization.

F. Deployment and Maintenance

After testing the product, it is deployed in its actual environment, where customer verifies his requirements, which is known as acceptance testing. After the feedback of customer, further enhancement is done, if necessary.

Thus, the software development life cycle is all about.

1) Understanding the problem.
2) Decide a plan for solution.
3) Coding the planned solution.
4) Test the actual program.
5) Maintain the product

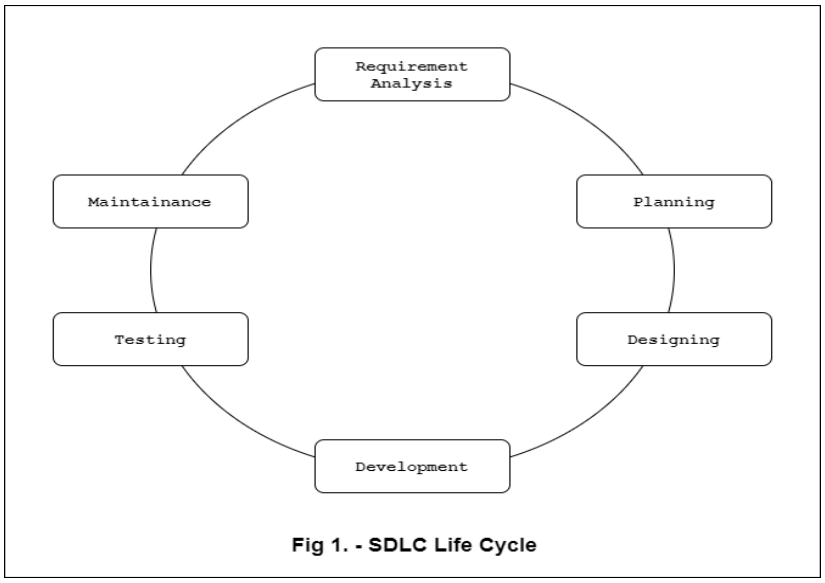


Fig 1. - SDLC Life Cycle

II. LITERATURE REVIEW

K Aggrawal, Yogesh Singh [1] gave a book named Software Engineering - 3rd Edition in 2007. This book gives a detailed account of all aspects of a software product, development failures and strategies to avoid them.

Barry Boehm,[2] in 2000 wrote a book on Development of Spiral Model, which was edited by Wilfred J. Hansen. This book elucidates about the spiral model's complete development and its various functionalities. Also providing a detailed description about its variants and invariants.

Vanshika Rastogi [3] gave a review paper named Software Development Life Cycle Models-Comparison, Consequences, International Journal of Computer Science and Information Technologies, Vol. 6 (1) , 2015. This paper is followed by discussion and comprehensive comparison among various SDLC models.

Nabil Mohammed Ali Munassar and A.Govardhan[4] have given a paper named A Comparison between Five Models of Software Engineering, published in Ijcsi International Journal of Computer Science Issues, Vol7, Issue 5,in Sep, 2010. It includes comparative study of various models, along with their strengths and weaknesses.

©IJRASET: All Rights are Reserved

Krishna, S. T., Sreekanth, D. S., Perumal, K., & Reddy, K. R. K [5], gave a paper named Explore 10 Different Types of Software Development Process Models, International Journal of Computer Science and Information Technologies, Vol. 3 (4), in 2012. This paper gave a summarized comparison of 10 types types of SDLC models, along with elucidating it's advantages and limitations.

Shaw, M. [6], gave a research paper named Writing good software engineering research papers, Proceedings. 25th International Conference on (pp. 726-736). IEEE in May, 2003. This paper clearly tries to lay the foundation of how one should go about writing lucid research papers in the field of software engineering.

SVITS, I. M.[7] gave a paper named A Comparative Analysis of Different types of Models in Software Development Life Cycle, International Journal, 2(5) in 2012. This paper gives a precise analysis of various software development life-cycle models, along with their comparison.

Sriramasundararajan Rajagopalan [8] gave a paper named Review on the myths on original software development model, International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.6, in November 2014. This paper emphasized on the use agile technologies. The understanding of this evolving type of model, with keeping the main focus is on the agile model.

III. BACKGROUND STUDY

The following are the models covered in this paper:

A. Waterfall Model

The Waterfall model was proposed by Winston W. Royce in 1970. This is one of the most classical models in software engineering. The Waterfall model is one of the oldest models, which is used in various government projects, as well is used by many big MNCs companies. This model is also known as the linear sequential model or classic life cycle. It has the following phases: Requirement Analysis, Design, Implementation, Testing and Maintenance.

All the above phases are fixed in such a manner that once done, they can't be repeated. As the Waterfall model is the classical model, all the other models have been derived from it in one way or the other.[1] The Waterfall model consists of several non-overlapping stages as shown in the figure bellow.

It is rightly said about this model that - "it is one way street with no turning back". You cannot move backward once phase "X" is over, you can move to phase "Y". There is no provision of moving backward.

1) *Waterfall Model Application:* It is not possible for one SDLC model to fit all the other types of software application. So, it is very essential to find the software model which best suits a specific software. The cases in which the Waterfall model is considered to be the best fit are as follows:

a) Where software requirements are well understood, documented, cleared, and static.
b) Software definition is stable.
c) Technology is accurately understood and is not dynamic (constantly changing).
d) There are no ambiguities in the listed software requirements.
e) The required resources are available.
f) The project at hand is small.

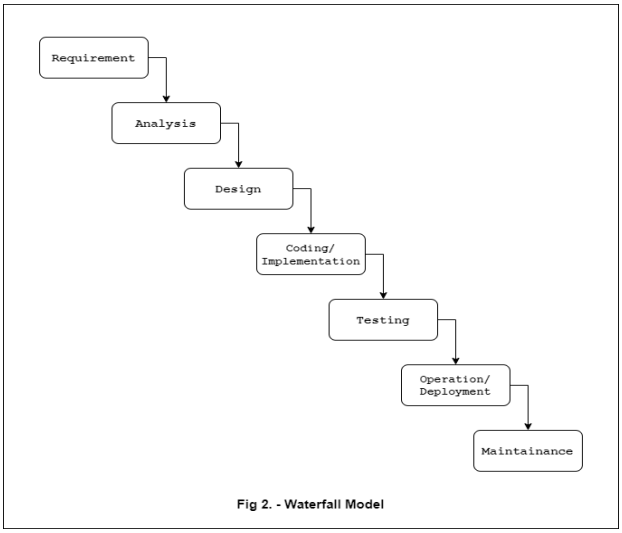


Fig 2. - Waterfall Model

©IJRASET: All Rights are Reserved

2) Advantages
a) All the requirements are defined before design.
b) It is used for projects which are simple and have strict deadlines.
c) Easy to understand.

3) Limitations
a) Idealized
b) Problems remain uncovered until testing.
c) Requirements are not clear.
d) Time consuming.
e) Not easy to integrate risk Management.
f) Difficult to make any changes
.

B. Spiral Model

The spiral model was developed during the late 1980s. It was outlined by Barry Boehm, and it introduced something crucial that all the other models failed to consider, which was risk analysis. In essence, the spiral model attempts to bring together key aspects of some other prominent models (namely the waterfall, incremental, and evolutionary prototyping), to gather the most appropriate traits from each one, as specific projects might be adaptable to specific models.[2] It has properties of all the models mentioned above. Spiral model is also known as meta model.[3]

1) *Spiral Model Application:* Spiral model is widely used in the software industry throughout the globe. The situations in which spiral model is best fitted are given below:
a) When constraints on budget and risk evaluation are important.
b) Medium to high complexity projects.
c) Where there is long term project commitment due to changing requirement environment.
d) When customers are not sure about all of their requirements.
e) Requirements are complex and need proper evaluation to attain clarity.
f) Where changes are expected in during life cycle.

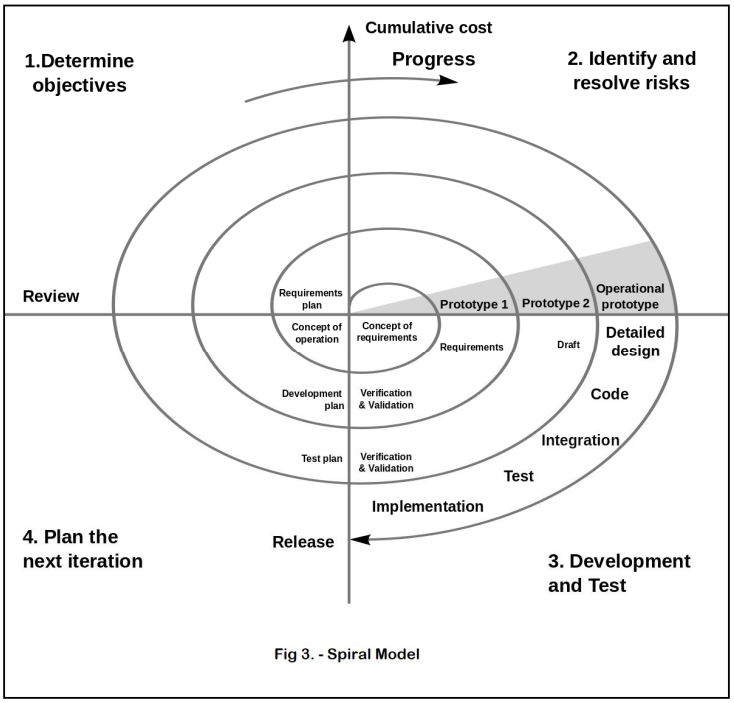


Fig 3. - Spiral Model

©IJRASET: All Rights are Reserved

2) Advantages

a) New prototype obtained every time.
b) Reuse capabilities.
c) Better productivity.
d) It has systematic stepwise approach.
e) Elimination of errors in early stage.

3) Limitations

a) Not proper cost and time estimation.
b) Cost of risk analysis is quite high on big projects.

C. Iterative Model

In iterative model, the complete requirements are not mentioned at the beginning and the process is initiated with a small set of requirements. Each iteration evolves a small version of product, and they are repeated until the desired version is attained.
Each iteration adds a new functionality in the process and continuous until the requirements are fulfilled.[4] The pictorial representation of iterative model is shown below.

1) Iterative Process Model Application: Like other SDLC models, Iterative model is not fit for every application.[5] However, this model is often used in the following cases:

a) When complete requirements of the system are neither clear, nor well defined.
b) Important and major requirements are definite; however, some functionality may evolve with time.
c) There is a time constrain.
d) A new technology is being used on the ongoing project.
e) Resources with needed skill set are not available. However, they are planned to be used on contract basis for some specific iterations.

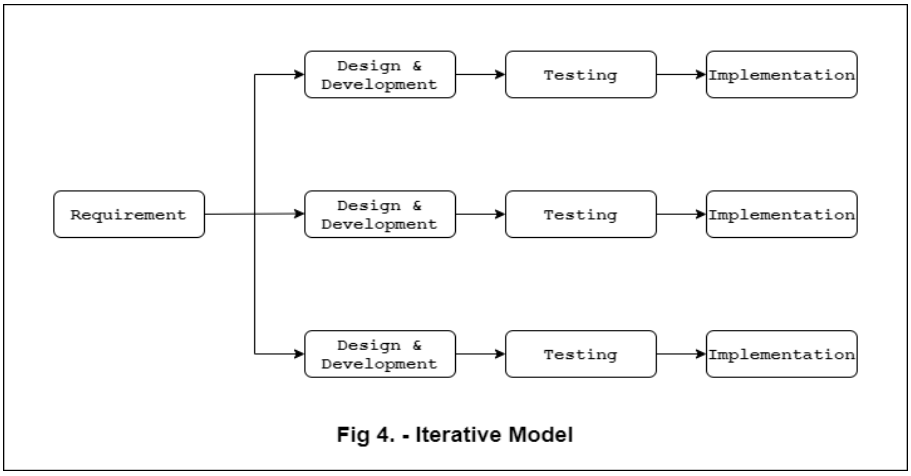


Fig 4. - Iterative Model

2) Advantages

a) Much better than other models for carrying out software processes.
b) Client feedback can be attained.
c) Can be used in those types of projects where requirements are not clear.
d) Document driven process.

3) Limitations

a) Not easy to manage.
b) Not clear milestones.
c) No stage is finished really.

©IJRASET: All Rights are Reserved

D. V-Shaped Model

The V-shaped model can be considered as an extension of the Waterfall model. In waterfall model we move in a linear way, whereas in V-shaped model, steps are bent in the upward direction from the coding phase to form a typical V-Shape.[6] There is a connection between each phase of the development process and its associated phase of testing.

The V-shaped model focuses more on testing, as testing is one of the most important steps in the software development process.

This model is almost similar to the Waterfall model, as both of them are sequential. Requirements must be very crystal clear before the project starts, as it can be very cost inefficient to go back to requirements and make the required changes for the existing project.[7]

1) V-Shaped Model Application

a) Where requirements are clearly defined and fixed.

b) When abundant technical resources are available with relevant technical expertise.

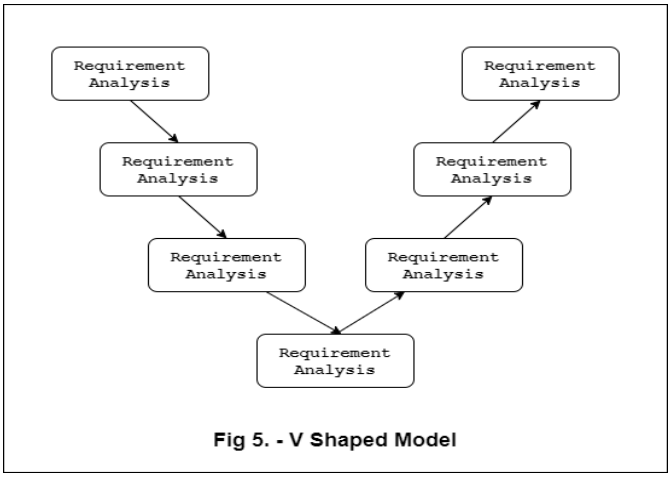


Fig 5. - V Shaped Model

2) Advantages

a) High amount of risk analysis

b) Good for critical projects.

c) Early production.

d) Easy to manage due to robustness of model.

e) Easy to understand.

3) Limitations

a) Not suitable for object-oriented projects.

b) Not suitable for projects with longer time duration.

c) Not suitable where requirements are fluctuating.

d) Can be quite costly at times.

e) Doesn't work well for small projects.

E. Rapid Application Development Model

The Rapid Application Development (RAD) model is based on prototyping and iterative development with no specific planning involved. The process of creating the software itself involves the planning required for developing the product.[8]

RAD majorly involves:

- Rapid Application Development focuses on gathering the client requirements through various workshops or focus groups.
- Early testing of the prototypes by the customer using iterative concept
- Reuse of the existing prototypes (components)
- Continuous integration and rapid delivery.

©IJRASET: All Rights are Reserved

1) RAD Model Application

This model is often used in the following cases:

a) When a system can be modularized to be delivered in an incremental manner.
b) When there is high availability of designers for modelling.
c) If the budget of the ongoing project permits to use automated code generating tools.
d) RAD SDLC model should be chosen only if domain experts are available with relevant and adequate business knowledge.
e) Where the requirements frequently change during the project.

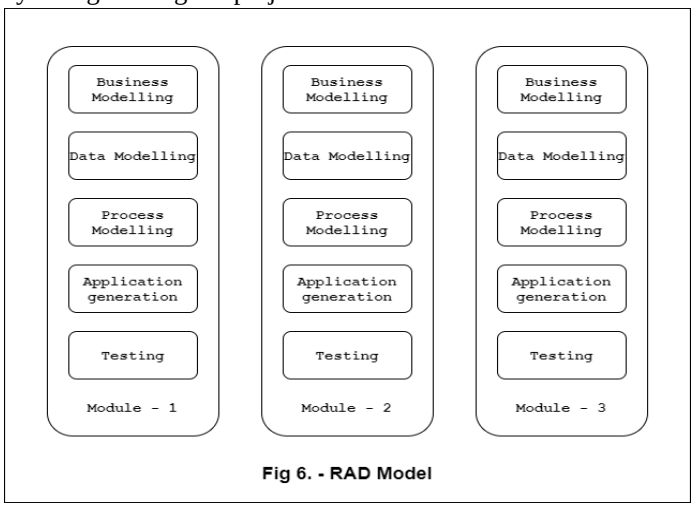


Fig 6. - RAD Model

2) Advantages

a) Changing Requirements can be accommodated.
b) Productivity with fewer people in a short time.
c) Reduced development time.
d) Increased usability.

3) Limitations

a) High Dependency on modelling skills.
b) Complexity of management is more.
c) Only the systems that can be modularized can be built using RAD.
d) Suitable for systems that are component based and scalable.

IV. CONCLUSIONS

This paper is about the different SDLC models and scenarios or situation in which these SDLC models are best used. It can aid project managers in deciding that what are the best suited models for their current project. Apart from big MNCs, these models help developers, teachers, student and any one for that matter, in resolving their software problems.

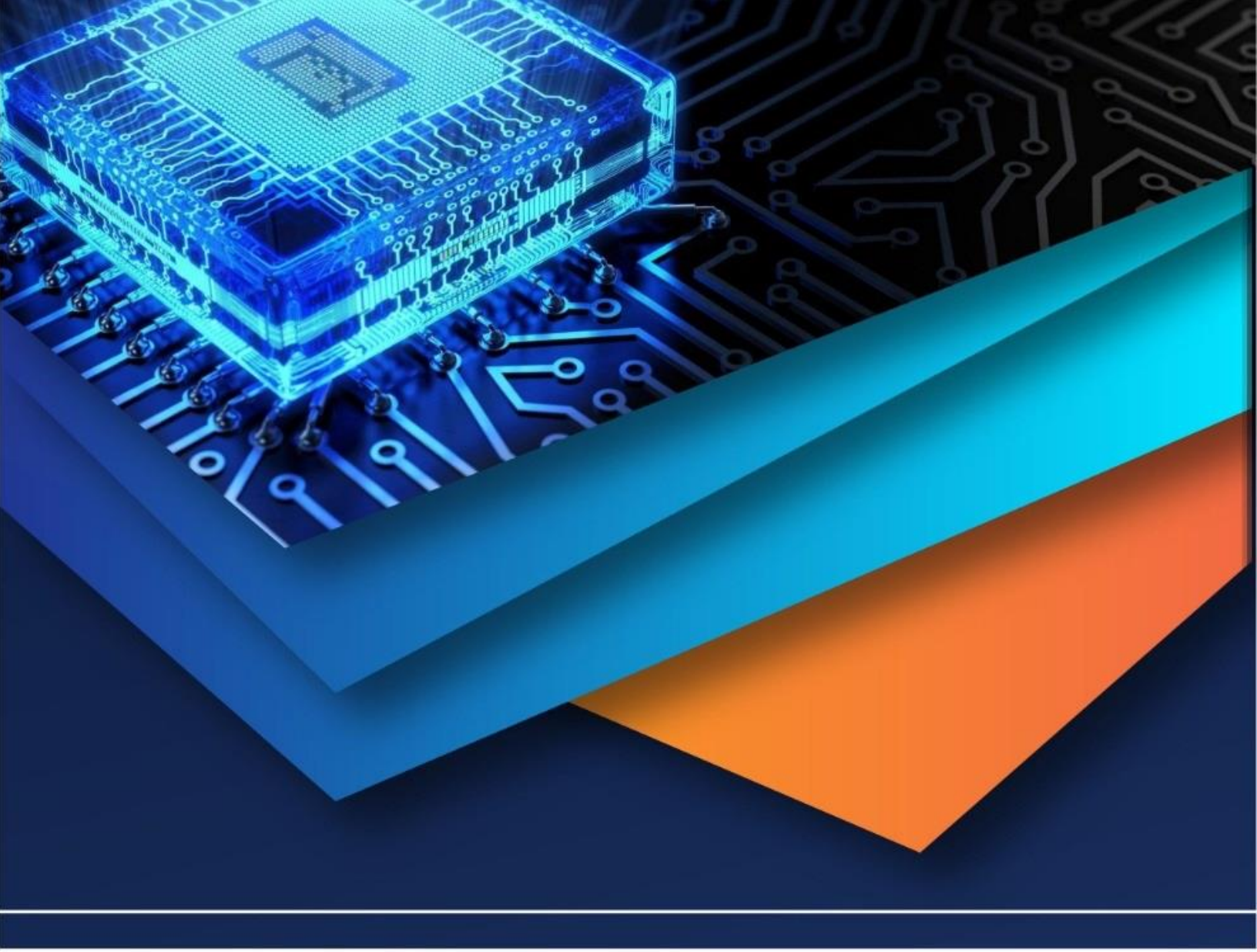
We have tried to incorporate almost all the popular SDLC models which are frequently used in the software industry. Waterfall and Iterative Model are the traditional and sequential models. Sequential refers to the process where the next phase will begin only after the completion of first phase. Such models are best suited for the projects where product requirements are crystal clear from the very beginning and where the requirements will not change dynamically during the entire course of project completion. Iterative and Spiral models, on the other hand, are more adaptive in terms of change and are suitable for projects where the requirements are not so well defined, or the market requirements change, more often than not. The V-model represents a development process that may be considered an extension of the waterfall model. In V-model, instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape. SDLC is an arbitrary approach to Software development and is suitable for smaller or academic projects. Rapid Application Development and Software Prototype are the advanced and innovative techniques to understand the requirements in a better way, early or beforehand in the project cycle. These techniques work on the concept of providing a working model to the customer and stockholders to give the look and feel and hence, eventually gathering their feedback. This feedback is used in an organized manner to improve the product and hence develop the best possible product.

©IJRASET: All Rights are Reserved

REFERENCES

[1] K. K. Aggarwal, Yogesh Singh Software Engineering 3rd Edition, 2007.
[2] Barry Boehm, "Spiral Development: Experience, Principles, and Refinements", edited by Wilfred J. Hansen, 2000.
[3] Vanshika Rastogi 'Software Development Life Cycle Models-Comparison, Consequences' ISNN: 0975-9646 International Journal of Computer Science and Information Technologies, Vol. 6 (1) , 2015, 168-172
[4] Nabil Mohammed Ali Munassar And A.Govardhan, "A Comparison between Five Models of Software Engineering" (IJCSI) International Journal of Computer Science Issues, Vol7, Issue 5, Sep, 2010
[5] Krishna, S. T., Sreekanth, D. S., Perumal, K., & Reddy, K. R. K. (2012). Explore 10 Different Types of Software Development Process Models. International Journal of Computer Science and Information Technologies, 3(4), 4580-4584.
[6] Shaw, M. (2003, May). Writing good software engineering research papers. In Software Engineering, 2003. Proceedings. 25th International Conference on (pp. 726-736). IEEE.
[7] SVITS, I. M. (2012). A Comparative Analysis of Different types of Models in Software Development Life Cycle. International Journal, 2(5).
[8] Sriramasundararajan Rajagopalan, "Review on the myths on original software development model" in International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.6, November 2014.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)