International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2390387

A Review on Software Development Life Cycle

Archi Agarwal, Avni Agarwal, Deepak Kumar Verma, Deepti Tiwari, Rashmi Pandey
Computer Science Department, Institute of Technology and Management, Gwalior, Madhya Pradesh, India

ARTICLEINFO

Article History:

Accepted: 01 June 2023
Published: 07 June 2023

Publication Issue
Volume 9, Issue 3
May-June-2023

Page Number
384-388

ABSTRACT

Efficiently developing dependable and high-quality software products within a strict timeframe should be a priority for those working within the technology sector. The Software Development Life Cycle (SDLC) process plays an integral role in this respect by offering various tried-and-tested methodologies such as Waterfall model Prototype model Spiral model Agile Incremental or Rapid Application Development model. Drawing from previous writings on this subject matter are used to investigate key SDCL principles outlined above.In this paper we are discussing about the SDLC phases and their different models.

Keywords: SDLC, Waterfall, Prototype and Agile, RAD.

I. INTRODUCTION

In fact we rely on them for 75% of our activities. To keep up with the ever increasing demands of the digital world we need computers - laptops or desktops - and software to utilize them effectively. All the functions in a computer operate based on the principles of its software. Thus, we require models to craft the essential software needed for creating other software.

II. SOFTWARE DEVELOPMENT LIFE CYCLE

One such model is SDLC as shown in Fig.1, which delves deep into the life cycle of software through various diagrams[1]. It covers all the steps that are vital for producing software products and moving them through their different stages until their retirement.

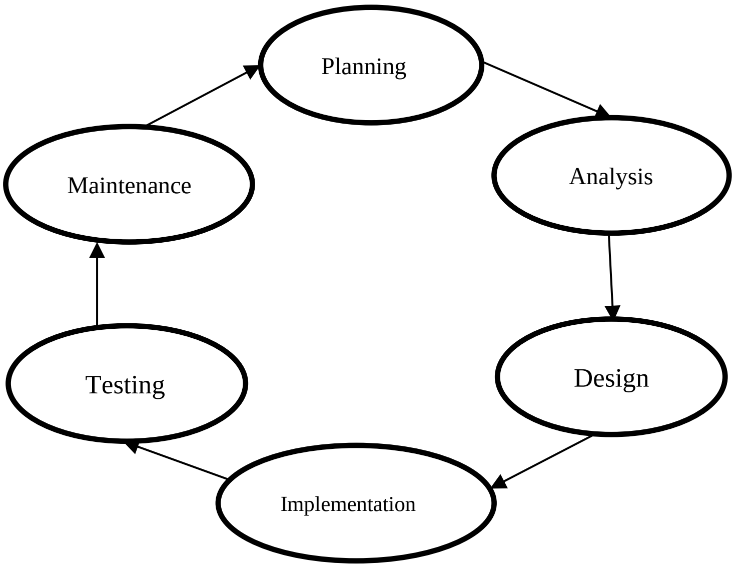

Fig.1. SOFTWARE DEVELOPMENT LIFE CYCLE

III.SDLC PHASES

Displayed in the cyclic diagram above are the following stages of SDLC. These stages comprise of:

1.REQUIREMENT ANALYSIS
2.DESIGN
3.IMPLEMENTATION
4.TESTING
5.EVOLUTION

When it comes to defining requirements for software the first step in the problem solving phase is a crucial one. After that the next phase is all about design. Once a solution has been identified its time for the coding phase where developers work on putting it into practice. The testing phase follows, where everything is put to the test - just to make sure all works as intended. And lastly, but definitely not least important we've got deployment and maintenance - essential stages that complete the lifecycle of software development process.

IV. SDLC MODEL

1. **Waterfall Model**— The model in question strictly follows a sequential approach where each phase must be accomplished before moving onto any subsequent ones [9][6]. Essentially, once a particular stage is concluded, its output effortlessly transitions or "flows" into its subsequent counterpart like a domino tile effect until all stages have been completed in order as intended. As shown Fig.2.

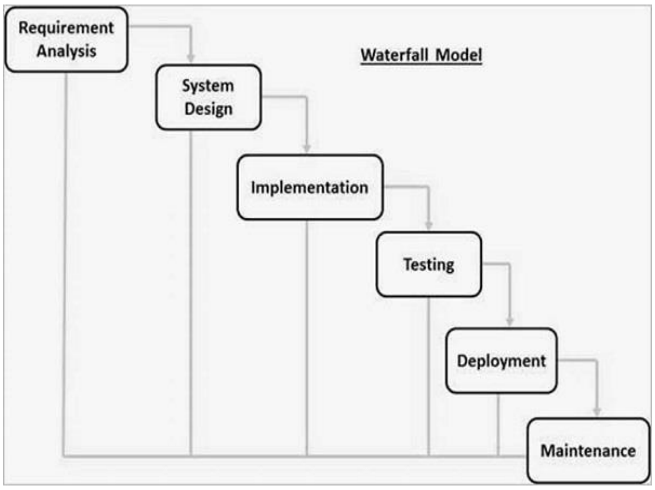

Fig.2. WATERFALL MODEL

2. **Prototype Model**- In order to comprehend how users think and receive their valuable input, developers utilize prototype models as part of the software development life cycle[8]. When requirements are vague or frequently changing, this approach proves useful. The process involves generating a functional version of the software that is later refined according to user feedback. Fig.3

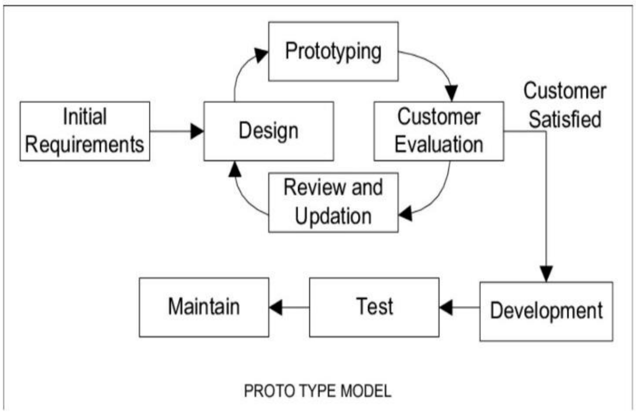

Fig. 3. PROTOTYPE MODEL

3. **Incremental Model**- The incremental model merges element of the linear sequential model and the iterative prototyping approach. The first increment of this model comprises a central product. However, even though it meets initial requirements, some auxiliary components remain absent[3]. Customers utilize the core product, after which a plan is devised for the next step

based on its results. Fig.4 shows the incremental model.

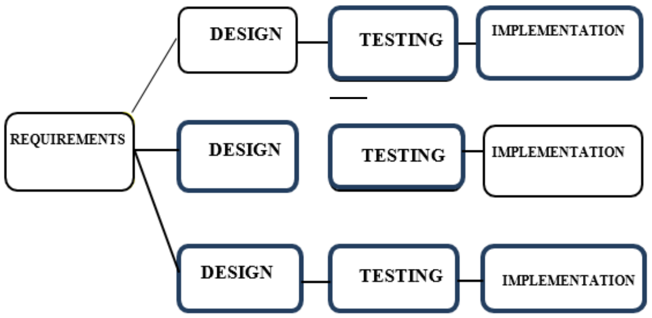

Fig.4. INCREMENTAL MODEL

4. Spiral Model-When it comes to software development models, there's one called the spiral model that's similar to the incremental model but has a greater focus on analyzing risks[4]. It consists of four main parts - planning, risk analysis, engineering and construction and release - that are repeatedly cycled through following various spirals. Starting with planning where requirements are collected; identifying any risks is addressed at this stage by providing solutions in subsequent phases such as engineering; producing tested software concludes this process followed by evaluation from customers. Fig.5

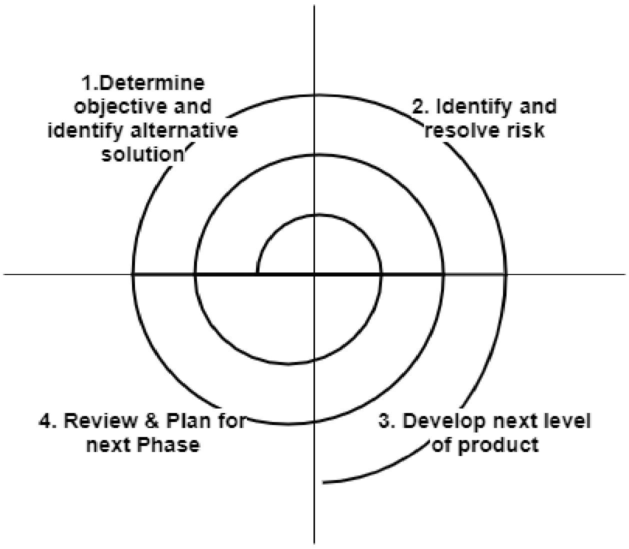
1.Determine objective and identify alternative solution
2. Identify and resolve risk
4. Review & Plan for next Phase
3. Develop next level of product

Fig.5. SPIRAL MODEL

5. Rapid Application Model (RAD)- The main objective is to enhance the quality of the outcome while reducing the production time. The key approach is to create components simultaneously and make prompt deliveries[2]. This allows customers to promptly utilize and request any necessary modifications. Fig.6 As a result , the team can deliver a functional model faster and in its entirety.

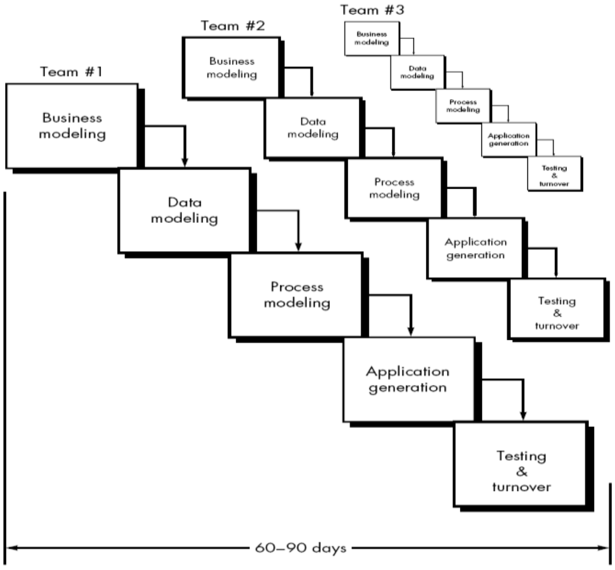

Fig.6. RAD MODEL

6.AGILE MODEL -Back in 2001, the agile team adhered to the agile methodology and presented this software, striving to meet clients' expectations by providing prompt and uninterrupted delivery of software [10][6]. Such a goal was accomplished primarily through generating minimalistic pieces of software step-by-step combined with speedy development cycles and amplified communication between both developers and clients as shown in the fig 7 . Moreover, the team demonstrated its adaptability when they faced any unexpected changes during the project implementation.

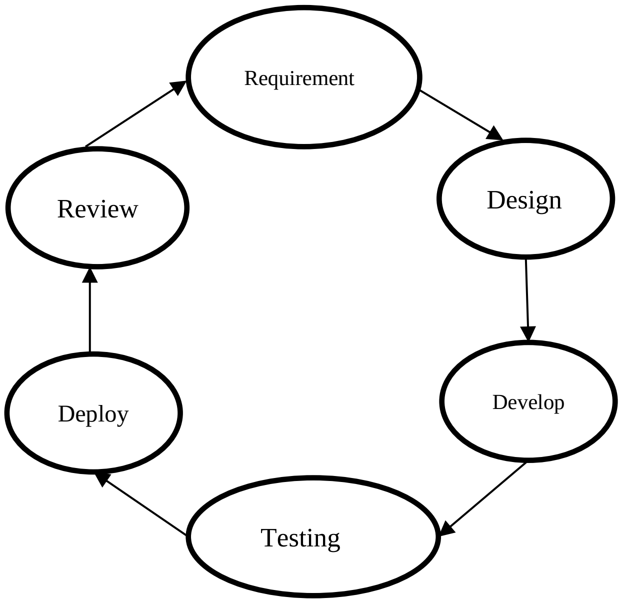

Fig.7 AGILE MODEL

V. CONCLUSION

We have read a number of publications on the SDLC, we discovered that there exist multiple methodologies for crafting software products. Each individual product carries its own merits and demerits; hence adopting specific models depends upon project requisites. But with the speed at which technology is developing today, it is more important than ever for developers to choose their preferred development model wisely in order to quickly provide high-quality products.

VI. REFERENCES

[1]. Christanto, H., & Singgalen, Y. (2023). Analysis and Design of Student Guidance Information System through Software Development Life Cycle (SDLC) and Waterfall Model. Journal of Information Systems and Informatics, 5(1),259-270. https://doi.org/10.51519/journalisi.v5i1.443

[2]. Arora, Ritika, and Neha Arora. "Analysis of SDLC models." International Journal of Current Engineering and Technology 6.1 (2016): 268-272.

[3]. Leloudas, Panagiotis. "Software Development Life Cycle." Introduction to Software Testing: A Practical Guide to Testing, Design, Automation, and Execution. Berkeley, CA: Apress, 2023. 35-55.

[4]. Dwivedi, Neha, Devesh Katiyar, and Gaurav Goel. "A Comparative Study of Various Software Development Life Cycle (SDLC) Models." International Journal of Research in Engineering, Science and Management 5.3 (2022): 141-144.

[5]. Garg, Apar, Rohit Kumar Kaliyar, and Anurag Goswami. "PDRSD-A systematic review on plan-driven SDLC models for software development." 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1. IEEE, 2022.

[6]. Sinha, Abhiup, and Pallabi Das. "Agile methodology Vs. traditional waterfall SDLC: A case study on quality assurance process in software industry." 2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech). IEEE, 2021.

[7]. Rajasekaran, N., and S. M. Jagatheesan. "Lack of SDLC Models and frameworks in mobile application development–a systematic literature review and study." Journal of Xi'an University of Architecture & Technology 13.8 (2021): 250-258.

[8]. Akinsola, Jide ET, et al. "Comparative analysis of software development life cycle models (SDLC)." Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1 9. Springer International Publishing, 2020.

[9]. Okesola, Olatunji J., et al. "Software Requirement in Iterative SDLC Model." Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1 9. Springer International Publishing, 2020.

[10].De Vicente Mohino, Juan, et al. "The application of a new secure software development life cycle (S-SDLC) with agile methodologies." Electronics 8.11 (2019): 1218.

[11].Anand Kumar Pandey, Rashmi Pandey, Neeraj Goyal (2022) “Performance Analysis of Multi

Agent System and Rule Based Expert System Design Approach Using Covid-19 Vaccination Process" The Design Engineering (Toronto) Scopus Indexed, Vol 2022 Issue 1, pp 3047-3053 April-22, ISSN: 0011-9342.

[12].Anand Kumar Pandey, Rashmi Pandey (2020) "Influence of CAP Theorem on Big Data Analysis" International Journal of Information Technology (IJIT), Vol 6, Issue 6, Dec 2020, ISSN: 2454-5414.

[13].Anand Kumar Pandey, Rashmi Pandey, and Ashish Tripathi (2020) "Underpinnings of Big Data Analytics and its Applications" International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT) Vol 6, Issue 2, March – 2020 ISSN: 2456-3307, DOI : https://doi.org/10.32628/CSEIT206222.

Cite this article as :

Archi Agarwal, Avni Agarwal, Deepak Kumar Verma, Deepti Tiwari, Rashmi Pandey, "A Review on Software Development Life Cycle", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 3, pp.384-388, May-June-2023. Available at doi :
https://doi.org/10.32628/CSEIT2390387
Journal URL : https://ijsrcseit.com/CSEIT2390387