International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

SOFTWARE DEVELOPMENT LIFE CYCLE MODELS: A REVIEW OF THEIR IMPACT ON PROJECT MANAGEMENT

Rajesh Goyal
FS - Insurance
IBM - US
Glen Mill, Pennyslvania, USA
Rajesh.nim@gmail.com

Abstract

This paper delivers a detailed evaluation of multiple SDLC models and their role in advancing project management. This investigation reviews classic systems such as Waterfall and V-Model alongside contemporary approaches like Agile, Scrum, and DevOps. The strength and weakness of each model are assessed according to important aspects of project management, including scope, time, cost, risk, quality, and communication management. The study's purview is to grasp the ways in which these models influence project success, efficiency, and adaptability within fast-changing technological surroundings

Keywords: SDLC, Project Management, Waterfall, Agile, Scrum, V- Model, DevOps

I. INTRODUCTION

The development of software is an ongoing process that has seen changes in the practices and frameworks that are used in the development of software. Software Development Life Cycle (SDLC) is the systematic approach that guides software engineers or project managers to develop, maintain, and even implement the software system in an efficient manner. It has a clearly defined process that shows all the core activities involved in the development of software include; requirement gathering, design, coding, testing, deployment, and maintenance [1].

Every SDLC model prescribes how to progress through these phases, providing varying levels of flexibility, documentation, testing, and risk management. Thus, the choice of an SDLC model greatly impacts the results of software projects: from traditional linear models, such as the Waterfall Model to the modern iterative Agile frameworks. Picking the appropriate model of SDLC, for project managers, is not just a technical decision but a strategic one; it determines resource intensity, risk management, and compliance with the scope, schedule, and funding are accomplished [2].

Project management in software development aims to deliver the project within the planned time frame, cost, and quality. The position of the project manager involves several aspects of the project such as the distribution of the tasks, the communication process, risk analysis, scope, and stakeholders. However, it should be understood that project management in different SDLC models should be conducted in different ways. For instance, Waterfall is a straightforward process with little flexibility as it progresses step by step; any change in the scope of the requirements will greatly affect its progress, while Agile is flexible; the project managers can easily adapt to changes in requirements and or market forces [3].

In the contemporary business world, organizations need to select the most appropriate SDLC model that will suit their project requirements, when addressing issues of change or dealing with emerging technologies. This decision often defines the success of a project because the choice of SDLC model influences the main areas of project management including scope, time, cost, quality, risk, and team communication [4].

This paper offers a comprehensive review of the key SDLC models such as Waterfall, V-Model, Agile, Scrum, and DevOps with special reference to Hermstein's choice theory and its applicability to project management practices. Thus, it can be concluded that the nature of project management considerations has fully complemented the advantages and limitations of SDLC models in order to achieve the best intended results to serve the purpose of the paper.

Further, it raises the question of how suitable these models are for current SW environments where requirements come from customers, and where markets and technologies are in constant flux. It is crucial to highlight that understanding the connection between the SDLC models and project management is essential for the enhancement of project success. It is with this understanding that the selection of an SDLC model can be the making or breaking of a project due to the ease of the implementation or complications that may arise in the course of the project hence distorting the planned goals of the project. Therefore, it is essential to present a comprehensive and profound analysis of the effects of the SDLC models on the critical factors of project management to assist the practitioners in making the appropriate decisions towards the achievement of the project objectives, the organizational culture, and the industry best practices.

II. LITERATURE REVIEW

The data presented in the study came from interviews with IT professionals who have worked in system development and IT project management. From a management standpoint, the objective of the study, Khatavakhotan , (2017) is to determine why an IT project failed. Information technology (IT) system development and IT project management are the subjects of the planned survey. According to the results, new IT models should prioritize time and quality management as the two most critical aspects of project management. The present paradigms of information technology development disempower software project managers by failing to establish a position of autonomous authority for these experts. Based on the results of the poll, it is clear that incorporating PERT and PDM into an IT development model helps project managers see the big picture and makes implementation easier. The findings highlighted the causes and requirements of the issue, and it is suggested that project tools be used in conjunction with the IT development process [5].

There are several models for software processes, albeit they are all somewhat narrow in scope. This study, Akbar, (2017) offers a new model for software development life cycles, the "AZ-Model," that incorporates additional activities into the process, based on this approach. As a result, it has a major influence on the timely manufacturing of a high-quality product while overcoming the shortcomings of conventional methods [6]

The Study, Dong, (2021) provides a concise overview of aerospace system engineering and aerospace software and then suggests a paradigm for rapid iterative development that relies on established technical solutions. New ideas for aircraft software development can be generated by applying this methodology to the creation of aerospace software products, which can simultaneously tackle efficiency and quality challenges [7].

According to, Chiang, (2020) Human resource allocation is crucial in project management, particularly in software development, for several reasons. Firstly, it helps software companies estimate how much projects will cost before deciding whether or not to contract them. Secondly, it ensures that projects are completed on time and of high quality. To reduce project time and costs while increasing efficiency and income, numerous methods have been suggested to aid in the allocation of human resources. In practical settings, these methods may fall short since they don't take into account important details that could affect the usefulness of an assignment, such as the cost of transfer, the quality of communication, and negative efficiency. To help software companies assess their current resources, decide if the tender estimate is feasible, and allocate human resources to form teams within fixed project durations, skill constraints, and budgets, this article proposes a framework. Using a real-world example as a foundation, software development project management aims to maximize performance while keeping costs and efficiency in mind. This approach comes close to meeting customers' expectations. Before assembling a team for a particular project with clearly defined timeframes and personnel assigned specific responsibilities, the values of critical framework factors are determined using individual computation foundations. Simulation research is carried out to show that the suggested model can help software organizations with their team formation decision-making process, and we formulate the decision-making process in integer programming models [8].

This paper, Unhelkar, (2009) proposes the "Composite Process Framework (CPF)" to address the lack of formality and rigidity in the existing software development processes and serve business needs more effectively. The paper concludes that reliance on a single software process does not work; rather, it requires a combination of factors derived from several lifecycles. The paper outlines a transition plan for organizations to execute the CPF with a focus on the need to customize the CPF framework more to the needs of the project. The CPF allows the successful working of different software projects that lead to the production of quality software applications that are delivered on time and within the estimated cost [9].

III. OVERVIEW OF SDLC MODELS

1. Waterfall Model

The Waterfall model is considered to be among the oldest and the most classical approaches used in software development, primarily due to the staged and sequential nature of the process that is inherent in the methodology. This model also organizes the development process into separate steps, which are the following: requirement analysis, system design, coding, integration and testing, and documentation and maintenance. The stages need to be fully executed before the next one can be carried out and they unfold in a cascading manner, hence the name. The Waterfall model stresses the methodical process which is characterized by the set stages, deliverables, and documented results at each phase. That is why it is useful in cases where the requirements are clearly defined at the beginning of the project and can hardly be altered during its implementation.

- **Impact on Project Management:** The Waterfall model seems to be quite easy to control from the point of view of a project manager since this model is strictly linear and the purpose and outcomes of each phase are predefined. This is particularly useful in tracking time and defining the scope because in each stage of project development, there is always some kind of understanding of what is expected to be delivered and within what period. The structured

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

timeline also allows for budgeting to be done in a more precise manner which makes resource management easier, especially when it comes to costs.

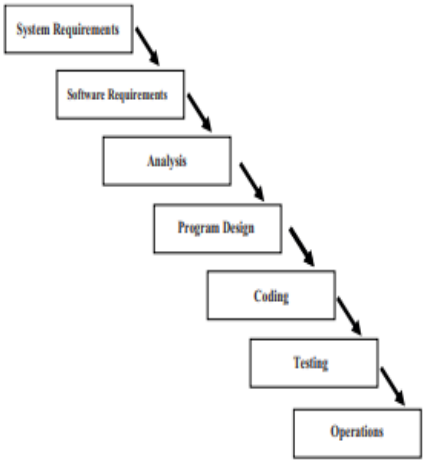

Figure 1: Waterfall Model [10]

They also ensure the proper process of documentation in responding to the questions of the stakeholders in every stage on the progress made on the particular project. Nevertheless, they also have some drawbacks when considering their flexibility in responding to the unpredictable changes in the requirements that can emerge at any stage of the Waterfall model. According to the developed plan on the project, once a phase is over it can be very expensive to make adjustments. For example, if new requirements are discovered during the design phase, one is compelled to return to earlier phases and as a result, time, cost as well as the project scope is magnified hence, the project is compromised. This makes the Waterfall model rather rigid and holding a higher risk in the scenarios where the flexibility or operation developed for the change is effective. Consequently, project managers can become frustrated by their organization's inability to factor changes in business or user requirements into the process, which can result in a higher likelihood of project failure in environments that are difficult to forecast or that change rapidly [11].

2. V MODEL

The V Model is recognized widely as the Verification and Validation model, and is an expanded version of the Waterfall methodology, which puts greater importance on the simultaneous planning of development and testing activities. This model connects each milestone of the project's growth with the appropriate time for testing, making sure that testing starts at the project's beginning. An illustration of the validation of system requirements is found in system testing, while integration testing is where the verification of design specifications takes place. This strategy leads to a "V" structure, where development stages flow down one side of the "V" and related validation activities take place on the other side. The main purpose of the V-Model is to enhance software quality by carrying out early and systematic testing at each stage of the software development process.

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

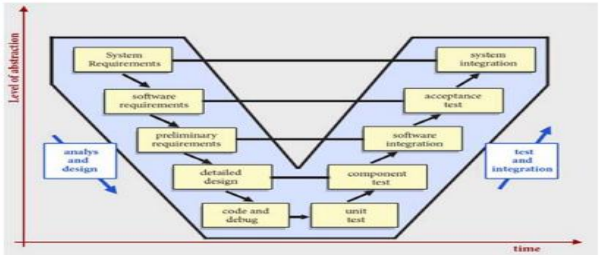

Figure 2: V Model [12]

- **Impact on Project Management:** A vital benefit of the V-Model for project managers is that testing is quickly integrated, prompting them to find flaws and errors early, lessening the likelihood that major problems will be found at the project's conclusion. Ensuring that quality assurance is implemented in every phase of development helps project managers reach better quality outcomes and reduces the need for costly rework at the close of a project. A strategic method to risk management can substantially lower long-term project costs and lead to better project dependability, helping to meet budget and time constraints [13]

3. AGILE MODEL

Agile technique development illustrates a pivotal gear shift from traditional linear approaches, including Waterfall and V-Model, by supporting an iterative and incremental strategy that nurtures adaptability in the face of change, cooperativeness with customers, and flexibility. Agile Trading practices replace a lengthy developmental cycle with fast, compact sprints that usually last 2 to 4 weeks each. Within each sprint, deliverables may comprise a fully developed product increment, thus enabling stakeholders to give ongoing feedback and allowing the development team to swiftly adapt to frequently shifting needs. The agile method relies mainly on teamwork and regular customer interactions to ensure that the finished product both fulfill client expectations and meets market needs. Highly regarded frameworks in the Agile sphere encompass Scrum, Kanban, and Extreme Programming (XP), all of which support iterative upgrades intended for the development of products for the customer.

- **Impact on Project Management**: By its dynamic nature, Agile provides project managers with the opportunity for better management of changing requirements and priorities. Project managers can monitor project progress routinely, given the sequential short sprints employed during the development process. An increase in adaptability strengthens risk management because it means detecting likely challenges or impediments sooner during development instead of towards the end. Also, ongoing interaction with clients contributes to better client satisfaction, because feedback is used continuously, making sure the desired requirements are met more efficiently in the final product. Agile allows project managers to adaptively prioritize tasks so that they can concentrate on offering the most critical features [14].

4. SCRUM MODEL

Scrum is considered one of the most extensively used frameworks in Agile methodology, created precisely to boost effective and adaptable software development through a very iterative process. Scrum is effective at simplifying complicated projects into smaller segments, managing them better using teams with multiple skills over short timeframes, dubbed sprints, which usually last 2 to 4 weeks. During each sprint, our purpose is to construct a specific element of the product that is

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

cohesive with our project ambitions and the desires of our clients. The cornerstone of Scrum are particular roles, such as the Scrum Master who handles the process and the Product Owner who manages the product backlog and prioritizes stakeholder items. Fundamentally rooted in practices, including frequent daily stand-up meetings (or daily scrums), Scrum facilitates the ability of team members to express their ongoing work, any barriers they encounter, and their future intentions. The team keeps a clear view of deliverables and is able to maintain the project's broader objectives by being transparent.

- **Impact on Project Management:** Scrum benefits include improved team effectiveness, as well as better communication, for project managers. The benefits of having a thorough list of sprint goals for the team are that they provide unambiguous targets for each iteration, which then enables the regular tracking of progress and identify any needed changes. The act of hosting stand-up meetings continually improves the way that team members interact with each other, giving project managers a chance to detect risks or constraints beforehand and take timely action by making necessary changes. Furthermore, the embedded nature of Scrum means that comment and enhancement can occur repeatedly, fostering better quality projects and accurately syncing client desires [15]

5. DevOps MODEL

DevOps is a fresh software creation process that focuses on merging the efforts of Dev and Ops teams, to simplify the complete delivery process for software from its earlier development until deployment and customer support. Rather than the traditional models, in which development and operations work separately, DevOps stimulates collaboration between these teams to achieve ongoing advances in development, integration, testing, and software deployment. Effective implementation of DevOps is highly dependent on automation, supporting methods like continuous integration (CI) and continuous delivery (CD) that automatically create updates to code, test them, and install these changes into production environments. Such an approach is predicated on feedback mechanisms that repeat, enabling fast detection and resolution of problems, subsequently improving both the timeliness and caliber of software releases. The basic premise of DevOps is fostering a culture that emphasizes collaboration and joint accountability, letting developers and operations function as a team, thereby enhancing responsiveness and agility to what clients ask for.

Figure 3: DevOps Mode [16]

- **Impact on Project Management:** For project managers, DevOps delivers important gains in terms of effectiveness and speed by getting rid of the traditional barriers that exist between

development and operations teams. Automated approaches combined with ongoing testing considerably trim the time necessary for software to make the transition from its developmental stage to deployment, therefore fostering more rapid updates and increased assurance. The results of more successful project outcomes are influenced by faster delivery cycles, less downtime, and greater software quality. The facilitative role of DevOps and its fundamental principles result in an expedient solution to issues, achievable through the involvement of multiple team members; thus, reinforcing project managers' skill in spotting and addressing risks from an early stage in the development. Enhanced speed of improvement is made possible by the real-time feedback and monitoring features of DevOps, while also boosting project agility and responsiveness to change [17].

IV. IMPACT OF SDLC MODELS ON KEY PROJECT MANAGEMENT AREAS

- **Scope Management:** Every approach to development management deals with scope management individualistically. The stable framework of Waterfall allows for the scope to be set crystal clear from the onset, while Agile allows for frequent redefined concepts of the scope. An important factor when determining a model is a project manager's ability to adapt to changing requirements.
- **Time Management:** SDLC models demonstrate a wide range of time management. Under stable conditions, both Waterfall and V-Model generally deliver effectively, improving the accuracy of timelines. The value placed on agile and scrum is their quick delivery nature, which regularly leads to faster releases but necessitates more flexible methods of time management.
- **Cost Management:** When evaluating it against Waterfall and the V-Model, the monitoring of costs is more closely supervised thanks to detailed planning that happens before the implementation phase. Still, the cyclic nature of Agile and Scrum can lead to an increased scope, complicating cost estimation. The addition of automation in DevOps helps cut long-term costs through an accelerated development and operations process.
- **Risk Management:** The essence of risk management is naturally embedded within each model. Waterfall and V-Model increase the possibility of higher risks if requirements change later in the project, whereas Agile and Scrum lower the risk by allowing small, incremental releases and consistent stakeholder feedback. Reducing risk is made possible by DevOps through continuous testing and monitoring.
- **Quality Management**: Various ways to manage quality are a unique gauge for evaluating SDLC models. In the development setting, both Agile and Scrum methods involve quality control, making them different from the Waterfall and V-Model methodologies, which prefer continual testing stages that need constant feedback and enhancements. Enhanced product quality is achievable through continuous integration and Testing supported by DevOps.
- **Communication Management**: Traditional models are contrasted by those like Agile and Scrum, which generally progress communication by promoting teamwork and daily stand-ups. In contrast, waterfall models are geared toward more detailed documentation and established phases, which increases the worry of communication breakdowns in settings that change swiftly [18].

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

V. COMPARATIVE ANALYSIS OF SDLC MODELS

A comparison of the SDLC models shows that no single approach suits all circumstances. The suitable model is determined by things like project scale, intricacy, stakeholder participation, and the organization's capacity to tolerate risk and change [19]. In Table 1, a comparison of these models is made with respect to key project management factors.

Table 1: Comparative Analysis of SDLC Models

SDLC Model	Scope Flexibility	Time Estimation	Cost Control	Risk Mitigation	Quality Assurance	Communication
Waterfall	Low	High	High	Low	Medium	Low
V-Model	Low	High	High	Medium	High	Low
Agile	High	Medium	Medium	High	High	High
Scrum	High	Medium	Medium	High	High	High
DevOps	High	High	High	High	Very High	Medium

VI. DISCUSSION

The decision to use a particular Software Development Life Cycle (SDLC) model is a major one that has a significant influence on project management results. The predictable nature of projects in which requirements are fixed and won't evolve fits nicely with the Waterfall model, allowing for transparent organization and good monitoring of progress. However, these designing strategies challenge adaptability, causing them to be less effective in dynamic environments, where agility is crucial. In contrast to alternative techniques, Agile and Scrum are ideal for situations that are rapid and unexpected, given their repetitive interactivity and client feedback, which renders them perfect for situations requiring prompt modifications. Despite this, these techniques can produce difficulties such as task creep and complications in predicting both time and cost in projects that continue over an extended period.

DevOps currently offers a more unified and automated approach by integrating development and operational teams, thus raising productivity and accelerating deployment rates. Even with its obvious benefits in enabling collaboration and expediting the launch of the software, DevOps can often be a costly investment in respect to resources, as it commonly necessitates major cultural shifts, technical proficiency, and the acquisition of automation tools. Ultimately, all models' success emerges from the unique project environment, which implicates the project's dimensions, complexity, and the flexibility needed to cope with changing requirements.

VII. CONCLUSION

This paper discussed the effect various SDLC models have on project management. According to the results, the best SDLC model decision is influenced by the unique project circumstances, risk profile, and operational needs. For projects that meet clear specifications, conventional models perform better, whereas Agile, Scrum, and DevOps operate most efficiently in settings that showcase adaptability and a requirement for ongoing optimization. Future projects in research could look into alternative models that integrate the most beneficial elements of different software development life cycle approaches to maximize project success.

REFERENCES

1. R. Arora and N. Arora, "Analysis of SDLC Models," Int. J. Curr. Eng. Technol., 2016.
2. P. Ragunath, S. Velmourougan, P. Davachelvan, S. Kayalvizhi, and R. Ravimohan, "Evolving A New Model (SDLC Model-2010) For Software Development Life Cycle (SDLC)," Int. J. Comput. Sci. Netw. Secur., 2010.
3. M. Niazi, S. Mahmood, M. Alshayeb, A. M. Qureshi, K. Faisal, and N. Cerpa, "Toward successful project management in global software development," Int. J. Proj. Manag., 2016, doi: 10.1016/j.ijproman.2016.08.008.
4. F. Ridha and E. Hegarini, "Analysis of Maturity Level Project Management of Software Development In Scrum Framework: Case Research On Tribe Enterprise PT. XYZ," IT J. Res. Dev., 2020, doi: 10.25299/itjrd.2020.vol5(1).4662.
5. N. H. Taba and A. S. Khatavakhotan, "IT system development via project management: A phenomenological and statistical study," in 2017 IEEE International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2017, 2017. doi: 10.1109/ICSIMA.2017.8311995.
6. M. A. Akbar et al., "Improving the quality of software development process by introducing a new methodology-Az-model," IEEE Access, 2017, doi: 10.1109/ACCESS.2017.2787981.
7. C. Dong, H. Fang, H. Zhang, Y. Wang, and X. Ling, "Research on Software Life Cycle Model Suitable for Aerospace System Engineering," in Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2021. doi: 10.1109/ICSESS52187.2021.9522355.
8. H. Y. Chiang and B. M. T. Lin, "A Decision Model for Human Resource Allocation in Project Management of Software Development," IEEE Access, 2020, doi: 10.1109/ACCESS.2020.2975829.
9. M. Al-Maharmeh and B. Unhelkar, "Applying a Composite Process Framework (CPF) in real life software development project," in ITNG 2009 - 6th International Conference on Information Technology: New Generations, 2009. doi: 10.1109/ITNG.2009.145.
10. W. V. C.-R. Gate and undefined 2017, "The Waterfall Model and the Agile Methodologies: A comparison by project characteristics," Researchgate.Net. 2003.
11. N. Hidayati and S. Sismadi, "Application of Waterfall Model In Development of Work Training Acceptance System," INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf., 2020, doi: 10.29407/intensif.v4i1.13575.
12. G. Kumar and P. K. Bhatia, "Comparative analysis of software engineering models from traditional to modern methodologies," in International Conference on Advanced Computing and Communication Technologies, ACCT, 2014. doi: 10.1109/ACCT.2014.73.
13. S. Balaji, "Waterfall vs v-model vs agile : A comparative study on SDLC," WATEERFALL Vs V-MODEL Vs Agil. A Comp. STUDY SDLC, 2012.
14. S. Aftab, Z. Nawaz, F. Anwer, M. S. Bashir, M. Ahmad, and M. Anwar, "Empirical evaluation of modified agile models," Int. J. Adv. Comput. Sci. Appl., 2018, doi: 10.14569/IJACSA.2018.090641.
15. E. Çetin and P. Onay Durdu, "Blended Scrum model for software development organizations," J. Softw. Evol. Process, 2019, doi: 10.1002/smr.2147.
16. P. Jha and R. Khan, "A Review Paper on DevOps: Beginning and More To Know," Int. J. Comput. Appl., 2018, doi: 10.5120/ijca2018917253.

17. M. Zarour, N. Alhammad, M. Alenezi, and K. Alsarayrah, "A research on DevOps maturity models," Int. J. Recent Technol. Eng., 2019, doi: 10.35940/ijrte.C6888.098319.
18. S. R. Duclervil and J. C. Liou, "The study of the effectiveness of the secure software development life-cycle models in IT project management," in Advances in Intelligent Systems and Computing, 2019. doi: 10.1007/978-3-030-14070-0_14.
19. G. Gurung, R. Shah, and D. P. Jaiswal, "Software Development Life Cycle Models-A Comparative Study," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., 2020, doi: 10.32628/cseit206410.