

Analyzing Parameters on the Usage of Software Development Methodologies: Agile, Lean & Dynamic System Development

Purvi Sankhe¹, Mukesh Dixit²

¹Research Scholar, ²Associate Professor
Sanjeev Agrawal Global Educational University, Bhopal

¹purvisankhe@gmail.com, ²mukesh.d@sageuniversity.edu.in

Abstract – In the rapidly evolving landscape of software development, agility has emerged as a cornerstone principle in navigating the complexities of modern projects. Agile Software Development is a paradigm that prioritizes adaptability, collaboration, and iterative progress, challenging traditional, rigid development approaches. This paper examines and analyzes three prominent software development methodologies: Agile Software Development, Lean Development, and Dynamic Systems Development, with a focus on key parameters such as project requirements, user involvement, development team dynamics, type of project, and associated risks. Each methodology is introduced, providing insights into their fundamental principles and methodologies. The analysis systematically assesses the suitability and effectiveness of these approaches in various project scenarios, shedding light on their unique strengths and weaknesses. By scrutinizing how each methodology addresses specific project requirements, engages users, forms development teams, caters to project types, and mitigates risks, this paper aims to provide valuable insights to assist organizations in making informed decisions regarding the selection and implementation of the most appropriate development methodology for their projects.

Keywords: Software Development Models, Agile Process, High and Light weight models

I. INTRODUCTION

In the dynamic realm of software development, the choice of methodology can significantly influence the outcome of a project. Agile Software Development, Lean Development, and Dynamic Systems Development stand as pillars of modern software engineering, each presenting distinct approaches to managing the complexities and demands of software projects. The selection of an appropriate methodology is crucial, as it shapes how projects are planned, executed, and ultimately, how successful they are in meeting their objectives.

This paper embarks on a comprehensive analysis of three prominent software development methodologies: Agile Software Development, Lean Development, and Dynamic

Systems Development. We delve into the fundamental principles and practices that define each methodology and their respective impacts on project success. The analysis centers on key parameters critical for successful project management: project requirements, user involvement, development team dynamics, project type, and associated risks.

Agile Software Development, known for its iterative and collaborative approach, places emphasis on adaptability and responsiveness to changing project requirements [1]. Lean Development, inspired by lean manufacturing principles, focuses on eliminating waste and maximizing value, aiming for efficiency and continuous improvement. Dynamic Systems Development Method (DSDM) aligns with iterative and incremental development, emphasizing active user involvement and a strong focus on business needs.

By scrutinizing how these methodologies align with and address project requirements, engage users, structure development teams, accommodate project types, and mitigate risks, this paper aims to provide valuable insights. These insights will assist practitioners, project managers, and decision-makers in selecting the most suitable methodology for their specific projects, ultimately contributing to successful project outcomes.

II. AGILE SOFTWARE DEVELOPMENT

It is a method to project management that places a focus on adaptability, teamwork, and iterative development. It is a methodology that places a high priority on client satisfaction and adjusts as needs change over the course of development [1].

Agile development's guiding concepts include [2]

- Persons and interactions, rather than procedures and tools
- Functional software above thorough documentation
- Customer involvement during contract negotiations
- Adapting to change versus sticking to a plan

Each iteration of agile development entails planning, designing, coding, testing, and releasing a tiny, working piece of software. This approach is iterative and

incremental. This strategy enables ongoing input and progress throughout the entire process.

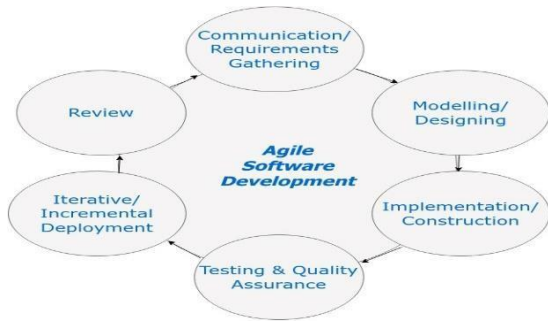


Fig.1- Agile Software Development Model

Agile development has a number of major advantages, including quicker delivery times, enhanced quality, more cooperation and communication, and greater adaptability to changing requirements. Agile, however, might not be appropriate for all projects or organizations, and for teams to really embrace it, there must be a fundamental transformation in mindset and culture [3].

Requirements of the project:

A product backlog, which is a prioritized list of features, additions, and bug fixes that need to be built, is how the requirements of a project are handled and established in the Agile process [3]. The product owner is in charge of establishing and prioritizing the requirements in accordance with the demands of the stakeholders and customers. The product owner is in charge of the product backlog [4].

Agile technique should have the following requirements-

Clarity and conciseness: Requirements should be written in a clear and concise manner to make them simple for the development team to understand.

Prioritized: The requirements ought to be ranked in order of the stakeholders' and clients' relevance and value.

Flexible: Agile methodology places a strong emphasis on the capacity to adjust to changing requirements; as a result, requirements should be adaptable and open to change as the project moves forward.

Testable: To ensure that they can be verified through testing and that the software satisfies the desired quality standards, requirements must be testable.

Measurable: The requirements should be measurable so that progress can be tracked and evaluated against the project goals.

User-focused: Agile methodology emphasizes delivering value to the customer, so the requirements should be user-focused and aligned with the needs of the stakeholders and customers.

User Involvement: User participation is a key component of the Agile methodology since it is necessary for producing software that satisfies the requirements of customers and stakeholders [5]. Agile encourages users to

participate in the development process by working regularly and continuously with the development team. Users are involved in Agile approach in a variety of ways, such as-

Participating in product backlog grooming sessions: To review and rank the project's needs, users are encouraged to take part in product backlog grooming sessions. Users are able to comment on the features and functionalities that are most significant to them thanks to this. Users are asked to attend sprint reviews to share their thoughts on the software that has been built during the sprint. Users can do this to view the project's development and offer suggestions for its course. Users are asked to take part in user acceptance testing (UAT) to test the software and make sure it satisfies their needs and requirements.

Providing feedback and suggestions: Users are encouraged to provide feedback and suggestions throughout the development process. This allows the development team to continuously refine and improve the software based on the needs and requirements of the users.

Development Team: In the Agile methodology, the development team is a self-organizing, cross-functional team that is in charge of gradually and iteratively producing the product. Developers, testers, designers, and other professionals who are required to deliver the software make up the development team [6].

The Agile methodology's development team should-

Be self-organizing: The development team should be given the freedom to organize and manage their work, including selecting the best course of action for handling tasks and managing the process.

Be cross-functional: The development team should have the knowledge and experience necessary to deliver the product gradually and iteratively, including responsibilities for software development, testing, and design as needed.

Be committed: In order to produce the software on schedule and within budget, the development team must be committed to the project.

Work cooperatively: To guarantee that the software satisfies the desired quality standards and that the project goals are achieved, the development team should collaborate with one another and with stakeholders.

Embrace continuous improvement: The development team should embrace continuous improvement and be open to feedback and suggestions to improve their processes and deliver higher quality software [7].

Be flexible: The development team should be flexible and adaptable to changes in requirements and project goals as the project progresses.

Type of Project and Risk Associated:

Software development, product development, marketing campaigns, and other projects that call for a flexible and iterative approach to development can all benefit from using the agile methodology [8]. However, some risks and

project types are particularly well-suited to the Agile methodology.

The following project types are appropriate for the agile methodology-

Agile methodologies are best suited for projects with changing or ambiguous requirements. Agile iterative and adaptable structure enables the development team to adjust as project needs change.

Projects with complex requirements: The agile methodology is well suited for projects with difficult to identify in advance complex or large-scale requirements [9]. Agile iterative structure enables the development team to divide complicated requirements into smaller, easier-to-manage portions.

High-uncertainty projects: Agile technique is appropriate for projects when it is challenging to forecast the ultimate result or deliverables.

Innovative projects: Agile methodology is suitable for projects that require a high level of creativity and innovation. The iterative approach of Agile allows the development team to experiment with different ideas and approaches, and quickly pivot if needed. The following risks are connected to the Agile methodology:

Absence of initial planning: Agile technique relies on adaptability and flexibility, which may result in absence of initial preparation [10]. This may lead to hazy project objectives, scope expansion, and a lack of a defined project direction.

Poor communication: The agile process calls for regular and honest dialogue between the development team and stakeholders. A lack of clear communication can result in errors, holdups, and subpar software.

Absence of documentation: Agile methodologies prioritize functional software over thorough documentation. However, this can result in a lack of documentation, which would make it challenging to support and maintain the product in the long run [11].

III. LEAN SOFTWARE DEVELOPMENT METHODOLOGY

By eliminating waste from the development process, lean software development is a project management strategy that puts the client first [13]. It is founded on the ideas of lean manufacturing, which place a strong emphasis on reducing waste in manufacturing operations.



Fig.2- Lean Software Development Model

Lean software development's fundamental tenets include-

Eliminating waste: Lean development seeks to get rid of any practices like overproduction, defects, over processing, waiting, excess inventory, superfluous motion, and untapped creativity that don't create value for the consumer. Lean development promotes ongoing learning and development through experimentation, introspection, and feedback.

Fastest possible delivery: Lean development seeks to release software as quickly as feasible by cutting down on the time needed for design, development, and testing. The significance of empowering the team to take charge of the development process and make decisions is emphasized by lean development [14].

Building quality in: Rather of depending on testing to find mistakes later in the development process, lean development prioritizes quality at every stage of the process. Faster delivery times, higher quality, greater customer happiness, and improved team morale are some of the major advantages of lean development. Lean, however, might not be appropriate for all projects or organizations, and for teams to fully embrace it, a considerable shift in mentality and culture is needed.

Requirements of the project: By removing waste, enhancing work flow, and continuously improving the process, the Lean Software Development technique aims to bring value to the customer. As a result, in lean software development, a project's requirements are centered on providing value to the customer in the most effective and efficient manner feasible [15]. The following are some essential specifications for a project in lean software development:

Customer value definition is the first step in the lean software development process. This entails establishing what the customer considers to be valuable and comprehending their needs, wants, and expectations. This aids in concentrating the project on giving the client the most crucial features and functions.

Waste identification is the second phase in the process. Finding inefficiencies and locations where work is not providing the consumer with value entails doing this. The development team may concentrate on providing value more effectively by avoiding waste.

Continuous development: Lean software development places a strong emphasis on this concept. This entails continually seeking out ways to enhance the procedure and get rid of waste [16]. All stakeholders should feel free to provide feedback and suggestions for improvement to the development team.

Enhance flow: The development team should attempt to improve how work is handled as it moves through the system. In order to do this, bottlenecks and delays must be kept to a minimum, and work must be processed quickly and effectively.

Empower the team: The development team needs to be given the authority to decide and handle their own job.

This entails granting them the power and resources necessary to provide the consumer with value.

Deliver incrementally: Lean Software Development emphasizes delivering value to the customer incrementally. This means breaking down the project into smaller, more manageable pieces and delivering them to the customer as soon as they are ready.

Ensure quality: Quality is a key requirement of Lean Software Development. The development team should work to ensure that the software is of high quality and meets the customer's needs and expectations.

User Involvement: Customer input and involvement are essential parts of the development process in lean software development. The methodology places a strong emphasis on the value of involving the customer throughout the entire development process to guarantee that the finished product satisfies their wants and expectations.

User interaction is emphasized in lean software development in the following ways-

Lean Software Development places a strong emphasis on ongoing client participation throughout the development process. This entails conversing with the client, obtaining their opinion of the software, and making any necessary corrections [17].

Iterative development: This method of developing software in small, incremental steps is a part of lean software development. This enables the client to comment on each iteration and modify the software as necessary.

User stories: The utilization of user stories is a crucial element in lean software development. User stories are succinct summaries of the software's features and functionality that customers want to see.

Collaborative development: Lean Software Development encourages collaboration between the development team and the customer. This can involve joint meetings, collaborative decision making, and other methods of working together to ensure that the software meets the customer's needs.

Development team: The development team is in charge of putting the Lean concepts and practices into practice in order to provide high-quality software that satisfies the needs of the client.

The development team in lean software development has the following important components-

Cross-functional team: In Lean Software Development, the development team often consists of people with a variety of backgrounds, skills, and specialties [18]. This can include participants in the development process such as programmers, designers, testers, project managers, and others.

Culture of collaboration: The development team is encouraged to cooperate and to openly exchange ideas and information. This makes it easier to make sure that everyone is working towards the same objectives and that everyone is aware of the aims of the project.

Improvement over time: The development team is also in charge of enhancing the development procedure over time. This may entail locating potential improvement areas, trying out novel methods and procedures, and making revisions as required. Implementing the iterative development approach, which is essential to Lean Software Development, is the responsibility of the development team. The development process is divided into manageable, incremental steps, and each step is improved upon until the finished product is delivered.

Customer focus: Lastly, it is the development team's job to keep the customer at the forefront of all decisions made during the development process. This entails communicating with customers frequently, taking their feedback into account during the creation process, and making sure the finished product lives up to their expectations.

Lean technique can be used on a range of projects, but it works especially effectively on those that are intricate, uncertain, or demand a lot of creativity. Because Lean methodology offers a flexible and iterative approach to development, it can help to manage the risk that is frequently associated with these types of projects [19].

The many project types and risks involved that the lean technique can be used for:

Software development projects: Software development projects can be quite difficult and uncertain, especially if they include creating new or cutting-edge software. By offering a flexible and iterative approach to development that enables the development team to modify the project as they get feedback from the client, the Lean methodology can help to manage the risk in this situation.

Product development projects: Product development projects can also be very complex and uncertain, especially if they include creating a brand-new or cutting-edge product. By offering a flexible and iterative approach to development, which enables the development team to modify the product as they receive feedback from the client, the Lean methodology can help to manage the risk in this situation.

Business process improvement initiatives: Business process improvement initiatives can be very unclear and difficult, especially if they require putting new or creative processes into place. By offering a structured approach to process improvement and enabling the team to design and test new processes in a controlled and iterative manner, Lean methodology can in this situation assist in risk management.

Lean methodology's capacity to control risk by offering a flexible and iterative approach to development is, in general, its main advantage. As a result, the development team can modify the product as they receive client input and thus iteratively enhance it over time.

IV. DYNAMIC SYSTEMS DEVELOPMENT METHODOLOGY

The Agile project management methodology known as Dynamic Systems Development Methodology (DSDM) is centred on completing projects on schedule, within budget, and to the desired quality [20]. The development team and stakeholders should prioritize collaboration and communication in this gradual and iterative approach.

The fundamental tenets of DSDM consist of:

Keep your eye on the ball: DSDM places a strong emphasis on comprehending and delivering the business value that the project is meant to achieve. Deliver projects on schedule, within budget, and to the desired quality. DSDM places a high priority on timely project delivery.

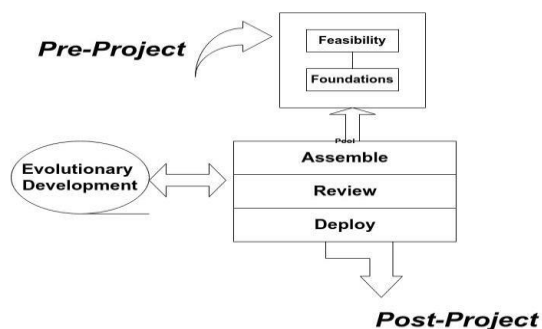


Fig.3- Dynamic Systems Development Model

Work together: Throughout the project lifecycle, DSDM places a strong emphasis on teamwork and communication between the development team and stakeholders.

Never skimp on quality: DSDM places a strong emphasis on preserving a high standard of quality throughout the development process methodology [21].

Build the project incrementally, starting with a strong foundation, and then adding functionality as the project moves forward. This is a key component of DSDM. Iterative development is a key component of DSDM, with each iteration yielding a functional, validated, and provable solution.

Continuous and transparent communication between the development team and stakeholders is stressed in the DSDM. Faster delivery times, stronger collaboration and communication, better quality, and greater adaptability to changing requirements are a few of the major advantages of DSDM. In order for teams to completely adopt DSDM, there must be a considerable change in thinking and culture, hence it may not be appropriate for all projects or organizations [22].

The goal of the agile development process known as Dynamic Systems Development process (DSDM) is to produce high-quality software quickly and affordably. DSDM emphasizes the value of providing a solution that satisfies the user's demands and is founded on the idea of user involvement. The project requirements are crucial to DSDM's ability to accomplish this goal [23].

The following are some important project requirements according to the Dynamic Systems Development Methodology:

Business needs: The identification and definition of business requirements is the first DSDM requirement. Understanding the project's aims and objectives as well as the business procedures that the software will support is necessary for this.

User requirements: Identifying and defining user requirements is the second DSDM prerequisite. This entails comprehending the requirements of the software's users, including their objectives, chores, and preferences.

Identification and description of functional needs are the third and final DSDM requirement. Based on the needs of the user and the organization, this involves determining the precise features and functionality that the software will offer. *Non-functional requirements:* The identification and definition of non-functional requirements is the fourth DSDM requirement. This includes elements that are crucial to the software's success, such as performance, scalability, security, and usability.

User Involvement: A fundamental tenet of the Dynamic Systems Development Methodology (DSDM) is user interaction. Users are the software product's ultimate beneficiaries, and DSDM is aware of how important it is for the project to have their feedback. Here are a few ways users participate in DSDM:

Gathering requirements: DSDM incorporates ongoing user participation in the requirements gathering process. The business, user, and functional requirements of the programme are identified and defined with the help of the users. This makes it easier to make sure the software satisfies user needs and is compatible with organizational goals.

Users are involved in the development process at every stage of the project because to the iterative nature of the DSDM methodology. At every level of the development process, users provide input on the product, ensuring that it is satisfying their needs.

Testing and acceptance: Users participate in the product's testing and acceptance. Users take part in testing and offer opinions on the effectiveness and usability of the product. This makes it more likely that the product will be high-quality and suit the needs of the user.

Users in DSDM are assigned distinct roles and duties inside the project. This comprises the "business ambassador" position, which stands in for the user base and makes sure that their demands are addressed, as well as the "tester" position, which offers input on the effectiveness and usability of the product [24]. User participation is, in general, a crucial component of DSDM. Users can help to guarantee that the software is high-quality and satisfies their expectations by being involved in the requirements collecting process, iterative development process, and testing and acceptance process.

Development Team: A development team in the Dynamic Systems Development Methodology (DSDM) normally

comprises of a small, cross-functional group of people who collaborate to produce a good or service. Typically, the following roles are present on the team:

Sponsor: The individual who supplies the initiative with funding and assistance.

Visionary: The person in charge of outlining and conveying the project's vision.

Corporate Ambassador: The individual who advocates corporate interests and offers advice on operational needs.

Technical Coordinator: The person in charge of organizing the project's technical components and assuring its viability from a technical standpoint.

Solution Developer: The individual in charge of creating, implementing, and testing the solution.

Solution tester: The person in charge of evaluating the solution to make sure it complies with the technical criteria and business needs.

Business Analyst: The person in charge of deriving functional requirements for the solution from the analysis of business requirements.

Team leader: The individual in charge of organizing and directing the development team.

Facilitator: The person in charge of facilitating meetings and inter-team and inter-stakeholder communication.

Projects requiring a high degree of adaptability and flexibility and that are complicated, uncertain, or call for the Dynamic Systems Development Methodology (DSDM), in particular, are ideally suited for this methodology. This method is frequently used for software development projects, but it can also be utilized for other projects where iterative development and ongoing stakeholder interaction are required. The potential for scope creep is one of the main dangers connected to DSDM. There is a chance that the project scope will go beyond the original needs due to the iterative nature of the process and the continued cooperation with stakeholders, which could result in delays and higher expenses.

There is also a chance that communication will break down when using DSDM. The strategy largely relies on stakeholders' and team members' excellent communication, and if it is not managed correctly, it can result in misunderstandings, delays, and quality problems [25].

The initiative could, at the very least, fail to produce the intended business benefit. This could happen if the project team is not focused on producing value at each iteration or if they do not have a firm grasp of the business requirements. It's crucial to comprehend the business objectives clearly and to order the needs according to their importance to the company in order to reduce this danger.

Table-1 Parameters to analyse software development Methodology selection

Parameters		Agile Development Methodology	Lean Development methodology	Dynamic System Development Methodology
Requirements of the project	Requirements are defined early in SDLC	Yes	Yes	Yes
	Requirements are easily defined & understandable	Yes	Yes	Yes
	Requirements are changed frequently	Yes	Yes	Yes
User Involvement	Requires Limited User Involvement	No	Yes	No
	User participation in all phases	Yes	Yes	Yes
	No experience of participating in similar projects	Yes	Yes	Yes
Development team	Little experience on similar projects	Yes	Yes	Yes
	Little domain knowledge new to technology	Yes	Yes	Yes
	Little experience on tools	Yes	Yes	Yes
	Training availability when needed	No	No	No
Type of project and risk associated	Project is the improvement of the old system	Yes	Yes	Yes
	Stable funding for the project	Yes	Yes	Yes
	Requirements are highly reliable	Cannot be determined	Cannot be determined	Cannot be determined
	Schedule of the project is tight	No	Yes	Yes
	Reusable component can be used	Yes	Yes	Yes
	Scare resources (time, money, people, etc.)	Yes	Yes	Yes

V. CONCLUSION

In conclusion, this research paper conducted a comprehensive analysis of three prominent software development methodologies: Agile Software Development, Lean Development, and Dynamic Systems Development Method (DSDM). Each methodology

offers distinct approaches to project management, emphasizing adaptability, efficiency, and user involvement. The paper focused on critical parameters such as project requirements, user involvement, development team dynamics, project type, and associated risks.

Agile Software Development, known for its iterative and collaborative approach, prioritizes client satisfaction and adaptability to changing project requirements. It offers advantages such as quicker delivery times, enhanced quality, and greater collaboration but requires a transformation in mindset and culture for successful adoption.

Lean Development, inspired by lean manufacturing principles, focuses on eliminating waste and maximizing value, aiming for efficiency and continuous improvement. It offers benefits like faster delivery times, higher quality, and customer satisfaction by delivering value effectively.

Dynamic Systems Development Method (DSDM) centers on completing projects on schedule, within budget, and to the desired quality through collaboration, iterative development, and continuous communication. It emphasizes user involvement to ensure high-quality software that aligns with user needs.

The research also presented a comparison of these methodologies based on key parameters, including requirements of the project, user involvement, development team dynamics, and project type with associated risks.

In selecting the appropriate methodology, project managers and practitioners need to consider the nature of the project, its requirements, the level of uncertainty, and the team's experience. Each methodology has its strengths and is well-suited for specific project types and circumstances. The decision should align with the project's goals and constraints, ultimately contributing to successful project outcomes.

Understanding and carefully choosing the appropriate methodology can significantly impact project success, aiding in efficient project planning, execution, and ultimately meeting project objectives

REFERENCES

- [1] Isaac Sacolick, Agile Methodology & Modern software development [online], (6 April 2022), Available at: <https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html>
- [2] P. Hohl, "Back to the future: origins and directions of the "Agile Manifesto" – views of the originators", *Journal of Software Engineering Research and Development* volume 6, Article number: 15 (2018). Cited on Oct 2023.
- [3] K. Singh, "Agile Methodology for Product Development: A Conceptual Study", *International Journal of Recent Technology and Engineering (IJRTE)*, Vol.10 Issue-1 ISSN: 2277-3878, May 2021.
- [4] Help with Agile & Scrum [online]. Available at: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/> Accessed on Sept 2023.
- [5] A. Mishra, S. Abdalhamid, D. Mishra, and S. Ostrovska, "Organizational issues in embracing Agile methods: an empirical assessment," *Int. J. Syst. Assur. Eng. Manag.*, vol. 12, no. 6, pp. 1420–1433, 2021.
- [6] R. Mokhtar and M. Khayyat, "A Comparative Case Study of Waterfall and Agile Management," *SAR J. - Sci. Res.*, vol. 5, no. 1, pp. 52–62, 2022.
- [7] Jana Pócsová *, Dagmar Bednářová, Gabriela Bogdanovská and Andrea Mojžišová, "Implementation of Agile Methodologies in an Engineering Course", *Educ. Sci.* 2020, 10, 333; doi:10.3390/educsci10110333 Published: 17 November 2020.
- [8] Ms. I.G.U., D. Rathnayaka, Dr. BTGS Kumara, "A Review of Software Development Methodologies in Software Engineering", *IJARIE-ISSN*, Vol-6 Issue-4, pp (O)-2395-4396, 2020.
- [9] T. Thesing, C. Feldmann, and M. Burchardt, "Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project," *Procedia Comput. Sci.*, vol. 181, pp. 746–756, 2021.
- [10] Agile Software Construction [https://link.springer.com/chapter/10.1007/1-84628-262-4_9], pp 161–182, cited on October 2023.
- [11] A. Ali, M. Rehman, M. Anjum, "Framework for Applicability of Agile Scrum Methodology: A Perspective of Software Industry", (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 9, 2017.
- [12] Documentation_Strategies_on_Agile_Software_Development_Projects [online] Available at: https://www.researchgate.net/publication/263088093_Documentation_Strategies_on_Agile_Software_Development_Projects. Accessed on September 2023
- [13] Amit R; Nagaraj G Cholli, "Application of Lean Principles in Software Development Processes", *IEEE 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, 1109/CENTCON52345.2021.9687927, February 2022.
- [14] Principles-of-lean-documents [online] Available at: <https://www.mastercontrol.com/gxp-lifeline>. Accessed on September 2023.
- [15] Lean Documentation [online] Available at: <https://www.kellton.com/kellton-tech-blog/lean-documentation-imperative-you>. Accessed on Sept 2023.
- [16] Software models [online] (August 2023) Available at: https://www.researchgate.net/publication/226323265_The_Spiral_Model_of_Software_Development_A_Risk-Driven_Process_Model. Accessed on August 2023.
- [17] R. Sundar, A.N. Balaji, "A Review on Lean Manufacturing Implementation Techniques" Elsevier *Procedia Engineering*, Volume 97, 2014, Pages 1875-1885.
- [18] P. Rodríguez, M. Mäntylä, M. Oivo, L. Ellen Lwakatare, P. Seppänen, P.Kuvaja, "Chapter Four - Advances in Using Agile and Lean Processes for Software Development", Elsevier *Advances in Computers* Volume 113, 2019, Pages 135-224.
- [19] Pi-Yu-Lin, "The Academic Development Trajectories of the Lean Production Based on Main Path Analysis Method", https://www.mdpi.com/journal/processes/special_issues/manufacturing_in_dustry, 29 July 2022.
- [20] What is dsdm [Online] available at: <https://www.agilebusiness.org/business-agility/what-is-dsdm.html>. Accessed on Oct 2023.
- [21] Dynamic System Development Method. [Online] Available at: <https://www.nimblework.com/agile/dynamic-system-development-method-dsdm/> Accessed on Oct 2023.
- [22] Dynamic System Development Method. [Online] Available at: <https://www.indeed.com/career-advice/career-development/dynamic-systems-development-method>. Accessed on Oct 2023.
- [23] A.K. Nazir, I. Zafar, M. Abbas, "The Impact of Agile Methodology (DSDM) on Software Project Management", *Circulation in Computer Science International Conference on Engineering, Computing & Information Technology (ICECIT 2017)*, pp:1-6, March 2018.
- [24] A. Sani, "A Review on Software Development Security Engineering using Dynamic System Method (DSDM)", *International Journal of Computer Applications* 69(25):33-44, May 2013.
- [25] S. Coyle & K. Conboy, "A Study of Risk Management in DSDM", *International Conference on Agile Processes and Extreme Programming in Software Engineering XP 2009: Agile Processes in Software Engineering and Extreme Programming* pp 142–148.