International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Comparative Analysis of Agile Frameworks: Evaluating their Impact on Software Development Lifecycle Efficiency

[1]Sharad Jain, [2]Dr.Varsha Jotwani, [3]Dr. Sanjay Agrawal

Submitted: 05/11/2024 **Revised:** 18/12/2024 **Accepted:** 27/12/2024

Abstract-This study presents a comparative analysis of Agile frameworks, including Scrum, Kanban, Extreme Programming (XP), and Scaled Agile Framework (SAFe), to assess their effectiveness in enhancing the Software Development Lifecycle (SDLC). A mixed-methods approach, incorporating qualitative and quantitative evaluations, is utilized to measure key performance indicators such as lead time reduction, productivity increase, defect rate reduction, and scalability. The findings indicate that Kanban excels in throughput and workflow efficiency, while XP leads in velocity and technical excellence. Scrum fosters iterative progress and stakeholder collaboration, making it ideal for dynamic environments. SAFe, with its structured scaling capabilities, is best suited for large enterprises. This research provides organizations with a structured comparison to aid in selecting the most suitable Agile framework based on their specific project needs and operational constraints.

Keywords: *Agile frameworks, Scrum, Kanban, Lean, SAFe, Agile methodology, software development lifecycle, project management, efficiency, adaptability, team collaboration.*

I INTRODUCTION

The globalization of the economy has forced organizations to face several challenges in the most wide-ranging business areas to achieve the agility and efficiency needed to remain competitive and adapt quickly to market changes [1]. As a result of this process, project managers have been challenged to adapt their processes to improve efficiency in increasingly agile environments, thus emerging alternatives to traditional methodologies for the management and development of projects. The agile development was initiated through professionals connected to the software engineering area, aiming to minimize the risks associated with software development. The Manifesto for Agile Software Development was created in 2001 based on four fundamental values: (i) individuals and interactions over processes and tools; (ii) working software over comprehensive documentation; (iii) customer collaboration over contract negotiation; and (iv) responding to change over following a plan. Compared to the traditional model, agile methodologies work with short cycles or interactions, in which at the end of each step there is a deliverable product. Consequently, they provide faster changes that adapt to the current paradigm of technological evolution and high market competitiveness [2]. Agile practices have been highly successful in the corporate market, especially among small teams and projects [3]. However, its adoption in large organizational structures is often questioned, given the difficulties of managing independence among multiple teams, hierarchical pyramids that are inspired by non-agile models and the difficulties arising from a cultural heritage of the industrial era. This view is confirmed by [4], [5] who identified that the large-scale application and institutionalization of agile practices within companies that develop software is challenging since the adoption of agile practices in large teams and projects requires that agile principles be applied throughout the organization. Consequently, the introduction of agile practices challenges the existing practical structures and launches issues related to traditional roles, responsibilities, and expectations. In [6] it is argued that some decisions need to be made regarding the strategy of implementation of 17 Journal of Applied Sciences, Management and Engineering Technology, Vol 2, Issue 1, March 2021:16-29 agile methodologies, particularly in their expansion to other

[1]*Research Scholar, Department of Computer Science and Application Rabindranath Tagore University, Bhopal(M.P)*
[2]*Professor, Department of Computer Science and Application Rabindranath Tagore University, Bhopal(M.P)*
[3]*Professor, Dean of Academic and Research with NITTTR Bhopal*

organizational areas and restructuring of business processes. The success of agile methodologies in projects and small teams necessarily led to their adoption in new areas, with increasingly more companies applying agile practices in large-scale projects with teams involving hundreds of professionals, often geographically dispersed [7], [8]. As a consequence of this process, frameworks have emerged to manage large-scale agile projects, such as Large-Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), Scale Agile Framework (SAFe), among others. These frameworks were developed considering a great variety of agile practices and covering multi and large teams in which the agile principles are applied throughout an organization. However, the choice of the most suitable and appropriate agile methodology for an organization is a problematic task. The results of the study carried out by [9] indicate that professionals in the engineering field pointed out the lack of an evaluation model to perform a comparative analysis among the various large-scale agile frameworks that allow guiding the practitioners in this choice. This situation has provoked little reflective and unsustainable decisions, in addition, to have paralyzed some transformation initiatives for large-scale agile. In this sense, this study intends to identify the large-scale agile frameworks that organizations can adopt and performs a comparative analysis between them to highlight

Concept of Agile Methodologies

Businesses are of different types, aims, structures, and processes. For this reason, agile methodologies are of different types for different agility functions.

Scrum

Scrum is a simple, but very popular agile framework. It offers guidelines for managing and overseeing the software.Three major elements make up Scrum: roles, ceremonies, and artefacts. The product/project development team, the scrum master, and the product/project owner/stakeholder are the positions that make up the Scrum Process. The individual who determines the project's features, money, and priority is the project owner. Maintaining the Scrum principles and practices falls within the purview of the Scrum master, who serves as the manager or leader. The group of self-organising, cross-functional individuals that work on the product is made up of four to seven members. Sprint Planning is one of the ceremonies when the objectives and the Sprint Backlog are selected from the Product Backlog. The product backlog is an organised collection of the necessary features and functionality that have been identified and ranked by the product owner and stakeholders based on their relative importance. It takes the form of user stories. At the same time, user stories that have been selected and prioritised by the team for the upcoming sprint are contained in the sprint backlog. The artefacts provide burnout charts in addition to the sprint and product backlogs. Additional rituals include the daily scrum meeting which is a brief stand-up that involves participation by stakeholders and team members. Sprint Review is a survey meeting that involves the customer and team members and includes a product demo that was built during the sprint. During a Sprint Retrospective, colleagues talk about the issues they ran across in the previous sprint and how to prevent them in future ones [13].

Extreme Programming

Extreme Programming, commonly shortened as XP, is an agile methodology that enables the software development process by managing code changes. Extreme Programming mostly focuses on proactive and automated testing. Testing constitutes a fundamental element of Extreme Programming and commences at a foundational stage of development. Instant and succinct development cycles with incremental design and planning are a feature of the XP methodology. High levels of client interaction are required at every phase of project/product development, which is another essential component of XP. In addition, the code in XP is daily deployed and integrated continuously [14] It makes use of a lightweight methodology that is adaptable to any size of software. XP is built around the ideas of partnered programming, thorough code reviews, and clear code. Furthermore, it enables flexibility to modifications made to the project's specifications later on, improving software quality and responsiveness to client requests for modifications. It also mandates “Open-Workspace” and “Small-Release,” which are development processes carried out in customer presence on-site and the incremental release of validated outcomes following testing [15].

Kanban

The goal of the Kanban work-management system is to prevent the team from promptly completing administratively assigned tasks while also improving the work environment. Kanban makes it

possible to visualise tasks that are typically hard to justify in concrete terms. This approach breaks down the tasks into manageable parts and uses the Kanban board to visualise them, giving the team an enhanced comprehension of the ongoing project. As opposed to Scrum, which has time constraints [16]

Kanban limits the quantity of work completed at any one time, commonly referred to as limiting the work-in-progress (WIP) metric. Additionally, members of Scrum are allocated set duties, whereas members of Kanban work according to their preferences because there are no specialised jobs or obligations assigned to Kanban members [17]. In Kanban, the product owner handles project administration, as opposed to scrum, where the scrum master manages the project. The primary ideas of Kanban are: 1) Process visualisation with a Kanban board. 2) Limiting work being done at any one time to maintain the team's focus on one task at a time. 3) Controlling the workflow by estimating the amount of time needed to avoid wasting time. 4) Constant feedback is needed to raise the calibre of the product. 5) Software and project development that is iterative and continuous

Lean

Compared to Scrum or XP, lean agile software development has fewer restrictions on principles, procedures, or regulations, which gives it a more flexible technique. Lean is mostly made up of seven values. Other agile methodologies complement these ideals because lean focuses primarily on reducing waste in the software development process [18]. Removing pointless operations from the software/project development process is another goal of lean, which aims to deliver more value in a shorter amount of time. Because Lean has few guidelines, using specialised tools is the major way to optimise the development process. In opposition to XP, which prioritises the customer and developer by attempting to reduce any tension that may occur from competing goals. Following a top-down methodology, lean establishes ideals and principles for senior management inside a company or organisation and is utilised to optimise the entire organisation [19]. The fundamental seven values of Lean are as follows: Firstly, eliminate anything that doesn't provide value to the product that needs to be built. Second, use the user's input to inform every iteration that you learn from. Third, wait to make critical decisions on software development until you have access to as much information as feasible, which is typically later in the project. Fourth, get delivery out as soon as you can. Fifth, create a setting that will support the developers' success. Sixth, the consumer ought to be the main priority. Finally, adopt a comprehensive perspective and strive to grasp the business's flow and how the product fits into it [20]. Conclusively, in the modern world, the business arena faces several shifting needs. The four pillars of agile methodologies are customer collaboration, working software, interactive teams, and change-responsiveness, according to the Agile Manifesto. As a result, adopting agile development has become essential for numerous organisations. As agile is an iterative process, errors are rectified and enhancements are implemented as the process advances. Rapid delivery and client satisfaction are the main advantages of an agile approach. Thus, to reap this benefit, agile needs to be implemented throughout the various stages that have been covered, such as conception, construction, and so on.

SAFe

The Scaled Agile Framework (SAFe) is a structured methodology designed to help large organizations implement agile principles effectively across multiple teams and departments. Unlike traditional agile frameworks like Scrum and Kanban, which are optimized for small teams, SAFe provides a scalable solution for enterprises by integrating Lean, Agile, and DevOps practices. It enables organizations to improve collaboration, accelerate product delivery, and enhance business agility while maintaining alignment between strategy and execution. SAFe is built on a foundation of Lean-Agile principles, emphasizing customer-centric development, continuous improvement, and iterative product delivery. It introduces a hierarchical structure that organizes Agile practices across different levels Team, Program, Large Solution, and Portfolio—to ensure effective coordination and alignment across the enterprise. By adopting SAFe, organizations can enhance efficiency, reduce time-to-market, improve product quality, and foster a culture of innovation. The framework supports Agile Release Trains (ARTs), which synchronize the work of multiple agile teams, ensuring seamless collaboration and continuous value delivery. SAFe also incorporates Lean Portfolio Management (LPM) to align business strategy with execution, helping enterprises prioritize and manage investments effectively. With built-in support for DevOps and Continuous

Delivery, SAFe ensures that enterprises can quickly respond to market changes and customer demands. Overall, SAFe is a comprehensive approach that enables large-scale Agile transformation by providing structured guidance, proven practices, and governance mechanisms to enhance productivity, transparency, and organizational agility.

II REVIEW OF AGILE METHODOLOGY

Jana Pócsová et al. (2020) explored the integration of the SCRUM framework into the Mathematics 1 course to enhance the instructional process and student engagement. The study aimed to address the evolving needs of students and the changing demands of employers who prioritize both soft and hard skills. During the 2019–2020 academic year, a test group of students participated in the revised course to assess its effectiveness. Two questionnaire analyses and a comparison of learning outcomes were conducted to evaluate the approach. The results indicated that students found agile methodologies beneficial in optimizing classroom time. The paper also presented a detailed plan for incorporating agile techniques into teaching, along with personal experiences, the impact on students' test scores, and recommendations for future implementation.[20]

Gleiston Guerrero-Ulloa et al. (2023) conducted an extensive study on the role of Scrum methodology in the development of Internet of Things (IoT)-based systems. The research initially gathered 4,303 documents, which were filtered to 186 relevant studies, and after content analysis, only 60 were considered. These documents categorized various approaches for building IoT systems, including methodologies focusing on software development and the system life cycle (SLC). The study found that 42.1% of IoT systems were developed solely using Scrum, while 10.5% incorporated additional agile methods such as Rapid Prototyping, eXtreme Programming (XP), or Kanban. The researchers also evaluated automated frameworks, platforms, and tools that enhanced IoT development. The study provided a structured review of current methodologies and offered insights into how agile approaches contribute to IoT software development.[21]

Patricia Losana et al. (2021) examined the integration of Personas—a user-centered design methodology—into agile development. The research sought to determine the level of acceptance and integration of Personas in agile frameworks. The authors reviewed 28 relevant publications categorized by agile methodology type and conducted a systematic mapping study. They identified common integration tactics across agile frameworks, as well as challenges related to persona modeling and context representation. The study proposed a modification to shorten the time needed to develop preliminary Personas, addressing agile development's time constraints. Findings suggested an increasing interest in the use of Personas within agile methodologies, as reflected in the growing number of publications on the subject.[22]

Jason Leong et al. (2023) investigated the sustainability of project management through the combination of agile methodologies and product management. The research addressed the growing concern t**h**at traditional project management methods may struggle to keep pace with rapid digital transformation. The paper analyzed various agile requirements engineering approaches published in recent years and discussed their integration with conventional project management techniques. The study emphasized that while agile methodologies have been widely adopted since 2011, their coexistence with traditional approaches is essential for maintaining adaptability in the software industry. The findings suggested that combining agile with standard project management methods ensures long-term project viability.[23]

Sanjaya Chathuranga et al. (2023) explored the application of agile project management techniques in building construction projects, particularly in the design phase using Building Information Modeling (BIM) technologies. The study reviewed literature on agile adoption in the construction industry and found that traditional waterfall project management remains dominant. However, with the rise of BIM software solutions, construction firms have an opportunity to adopt agile approaches. The researchers examined a design company implementing agile project management and identified ten best practices for facilitating agile adoption. Techniques such as iterative design, continuous integration, cross-functional teams, sprints, and backlog maintenance were found to be effective. The study concluded with theoretical and practical recommendations for further research and agile adoption in construction projects.[24]

Daniel Soares et al. (2022) focused on the challenges of implementing agile project management in the automotive industry, particularly

in Portugal. The study addressed a gap in existing literature by investigating the difficulties businesses face when adopting agile methodologies. A survey was conducted with 56 fully completed responses from 148 automotive component manufacturing companies. Statistical analyses using Kruskal-Wallis, Mann-Whitney, and Spearman's correlation tests were performed to evaluate the findings. The study identified key barriers to agile adoption, categorized as organizational, knowledge and technology, institutional, and financial obstacles. The authors proposed a framework to overcome these challenges, emphasizing the importance of agile values, initial investments in skilled teams, and the adaptation of workflows based on company-specific factors. The study suggested that these insights could be applied to other countries for comparative analysis.[25]

Fernando Almeida et al. (2024) analyzed the challenges of scaling agile methodologies in enterprise settings where large-scale, multi-stakeholder projects are involved. The study reviewed 1,509 papers using the PRISMA 2020 methodology and identified 38 relevant studies. The findings highlighted that most studies on agile scaling have emerged since 2021, with case studies being the predominant research approach. The research explored hybrid agile scaling methods that integrate agile with traditional project management techniques. The concept of ambidextrous strategy—balancing flexibility with structure—was found to support hybrid agile scaling. The study provided valuable insights into how organizations can tailor agile methodologies to their specific needs and outlined best practices for overcoming challenges in large-scale agile implementation.[26]

III RESEARCH METHODOLOGY

The proposed approach focuses on analyzing and enhancing Agile software development frameworks by integrating modern methodologies, tools, and performance evaluation techniques. It involves selecting and categorizing agile frameworks such as Scrum, Kanban, SAFe, and Lean, followed by a comparative analysis based on flexibility, scalability, collaboration, and efficiency[27]. Key performance metrics, including sprint velocity, lead time, defect density, and customer satisfaction, is defined to evaluate effectiveness.

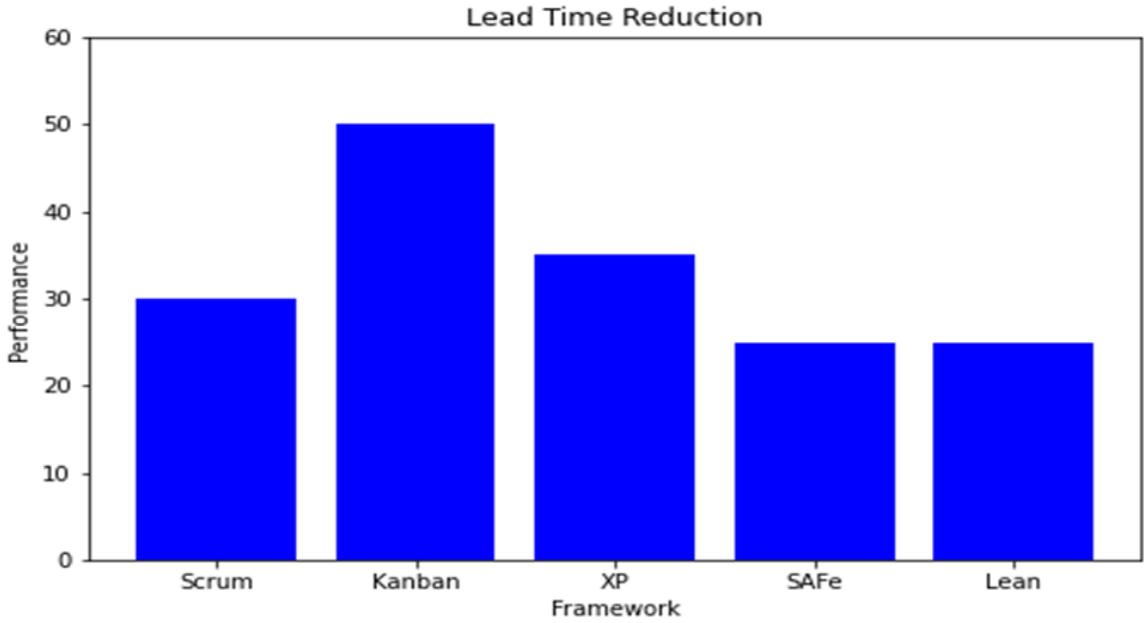
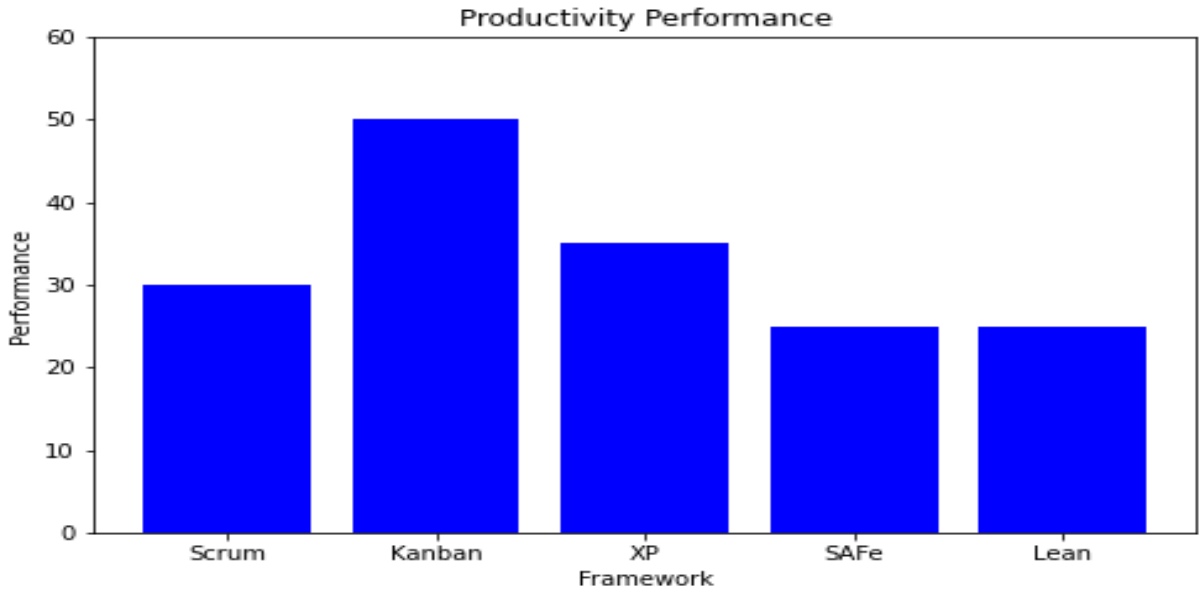


Fig. 1 lead time reduction

This figure shows the percentage reduction in lead time for each Agile methodology. Lead time represents the total time taken from the initiation of a work item to its completion. Kanban demonstrates the highest lead time reduction (50%), followed by XP (35%) and Scrum (30%). SAFe and Lean have the lowest lead time reduction (25%), indicating their focus on structured workflows over rapid task completion.



ig.2 productivity performance

This figure highlights how each Agile methodology improves productivity by measuring the percentage increase in output. XP (35%) and Kanban (50%) show significant productivity improvements, reflecting their iterative and continuous delivery approaches. Scrum and SAFe both show moderate gains (25%), while Lean (25%) emphasizes steady but efficient production.

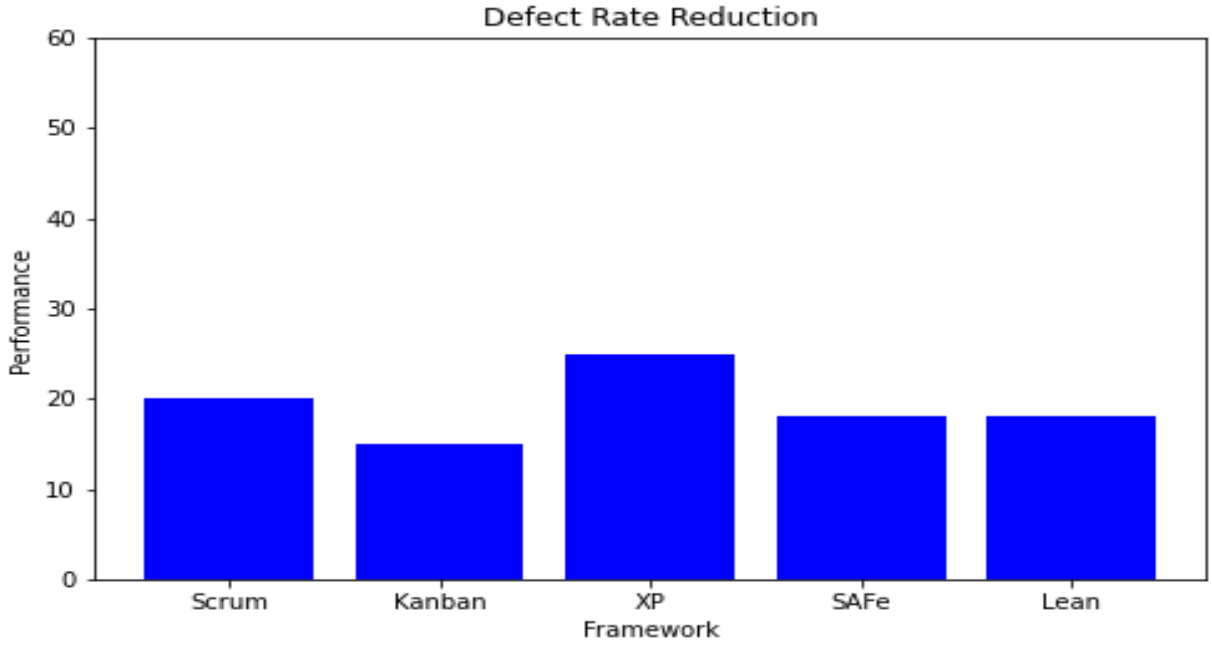


Fig. 3defect rate reduction

This figure illustrates how effectively each methodology reduces defects in software development. XP (25%) leads in defect reduction due to its strong emphasis on Test-Driven Development (TDD) and pair programming. Scrum and SAFe achieve moderate defect reductions (20% and 18%, respectively), while Kanban (15%) and Lean (18%) focus more on continuous process improvements rather than strict defect prevention strategies.

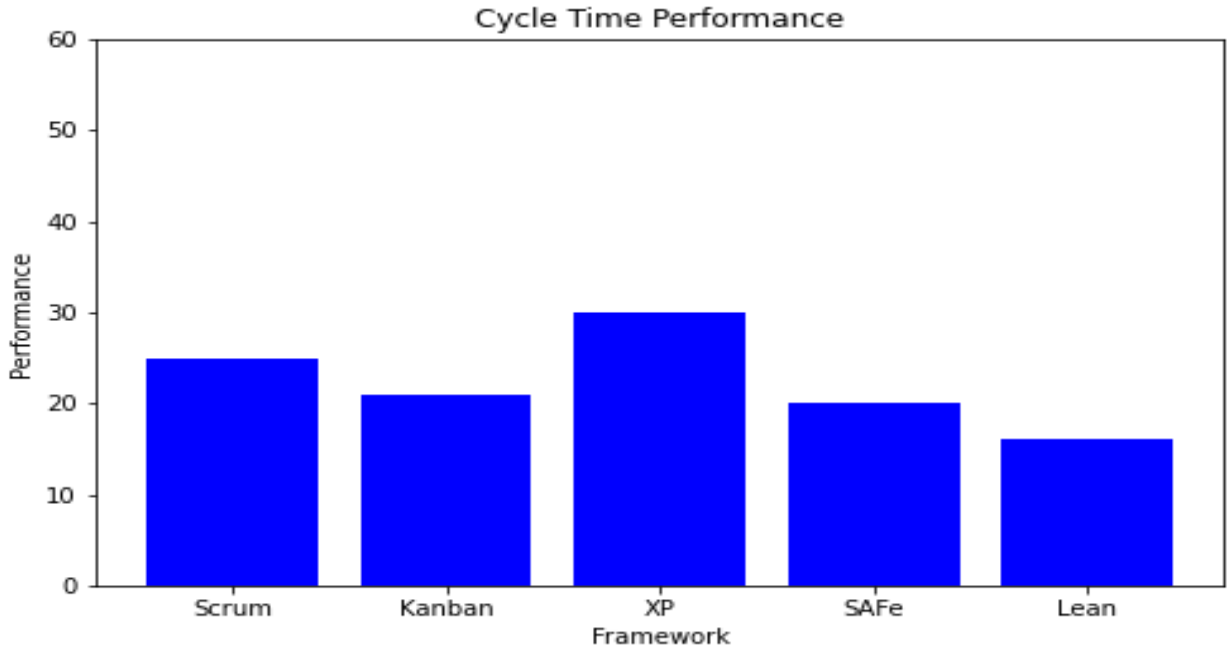


Fig. 4 cycle time performance

This figure displays the impact of each methodology on cycle time—the time taken to complete one work item. XP (30%) shows the highest reduction in cycle time, aligning with its fast-paced iterations. Kanban (21%) follows, reflecting its ability to optimize workflow efficiency. Scrum (25%) and SAFe (20%) also provide structured improvements, while Lean (16%) maintains steady cycle time management.

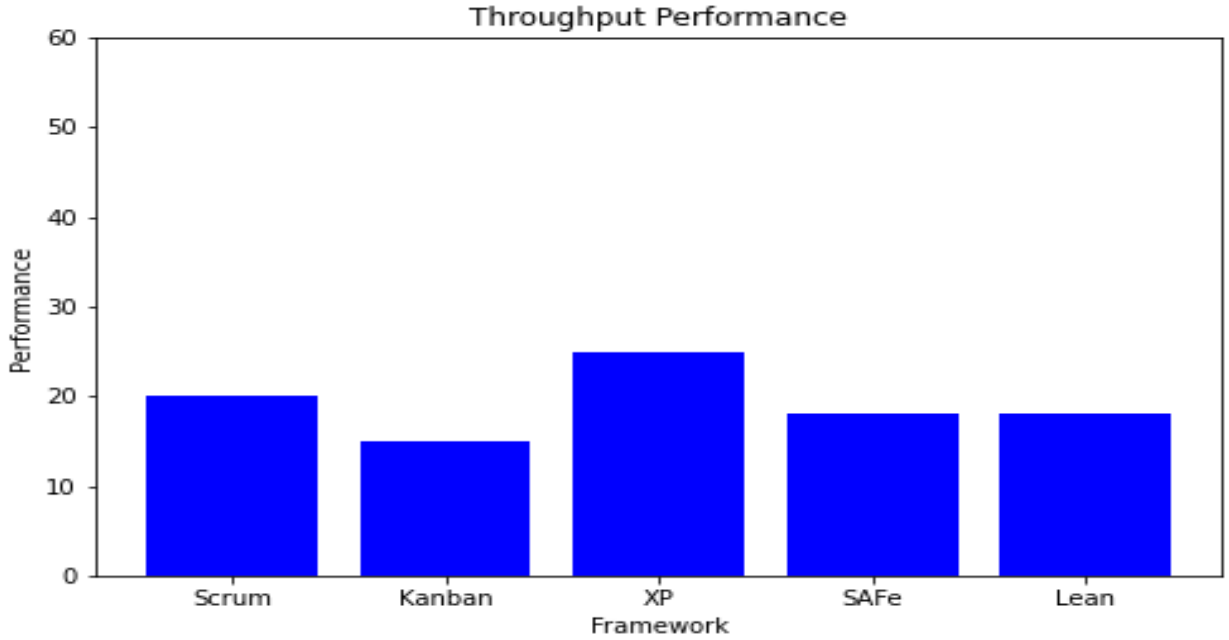


Fig. 5 throughput performance

This figure represents the number of completed work items within a given timeframe. XP (25%) and Scrum (20%) show strong throughput performance, emphasizing high team velocity. Kanban (15%) focuses on continuous flow, which leads to steady but slightly lower throughput. SAFe (18%) and Lean (18%) maintain moderate throughput by balancing efficiency and scalability.

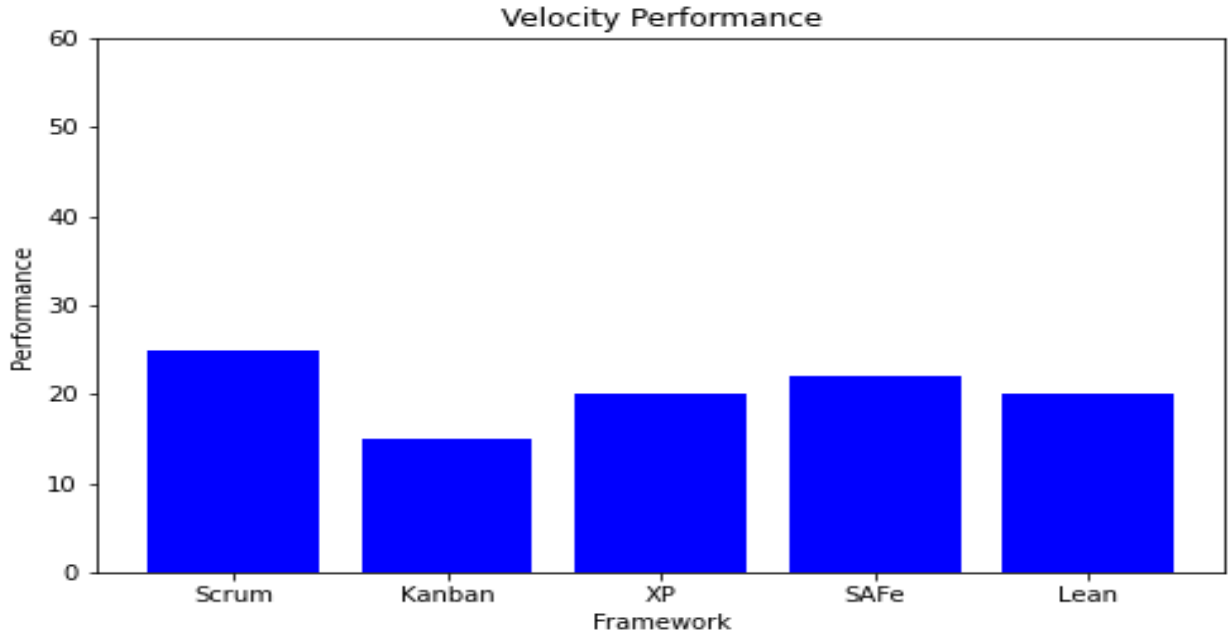


Fig. 6 Velocity performance

This figure illustrates team velocity, representing the number of tasks or story points completed per sprint. Scrum (25%) and XP (20%) exhibit strong velocity due to their structured sprints and iterative approaches. Kanban (15%) focuses on continuous delivery rather than sprint-based velocity. SAFe (22%) provides enterprise-wide velocity, while Lean (20%) emphasizes efficiency-driven velocity improvements.

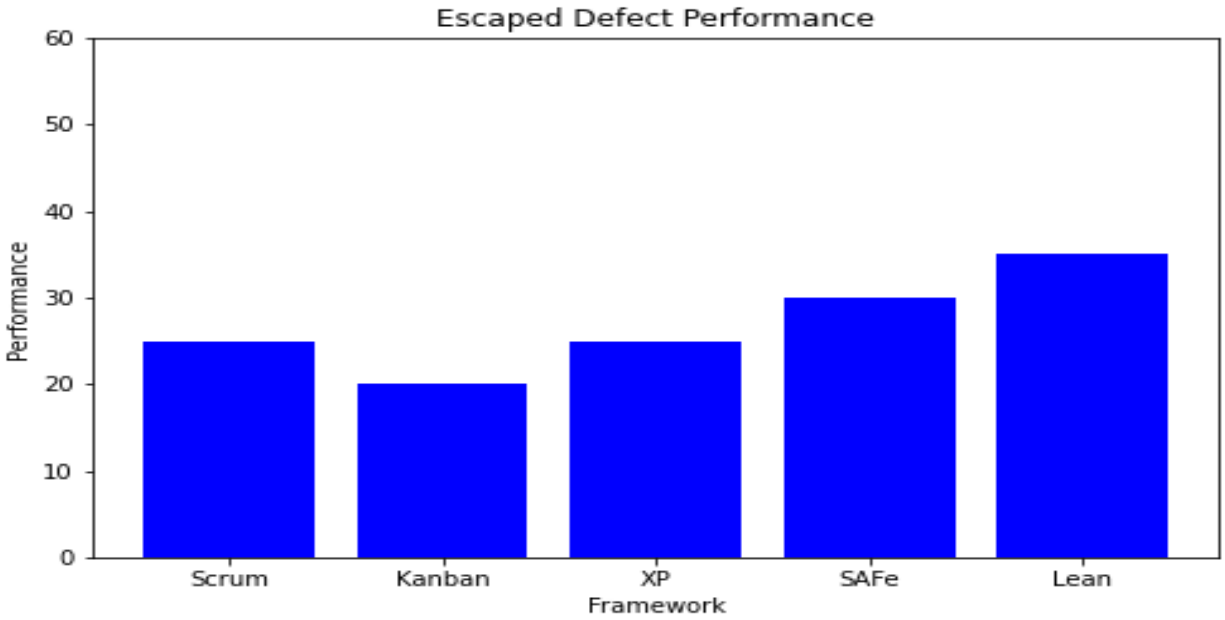


Fig. 7 escaped defect performance

This figure shows how many defects escape into production despite testing. Lean (35%) and SAFe (30%) demonstrate strong performance in reducing escaped defects, ensuring high-quality releases. XP (25%) also performs well, leveraging rigorous testing practices. Scrum (25%) maintains balanced defect management, while Kanban (20%) focuses more on workflow efficiency than defect prevention.

Table 1 Comparative Metrics across Agile Methodologies

Framework	lead time reduction	Velocity performance	productivity performance	cycle time performance	throughput performance	defect rate reduction	escaped defect performance
Scrum	30	25	30	25	20	20	25
Kanban	50	15	50	21	15	15	20
XP	35	20	35	30	25	25	25
SAFe	25	22	25	20	18	18	30
Lean	25	20	25	16	18	18	35

Table 1 provides a comparative analysis of Agile methodologies based on key performance metrics. Kanban demonstrates the highest lead time reduction (50%) and productivity performance (50%), making it ideal for continuous workflow optimization. XP excels in defect rate reduction (25%) and cycle time performance (30%), reflecting its strong emphasis on Test-Driven Development and iterative refinement. Scrum balances productivity (30%) and velocity (25%), ensuring structured teamwork and incremental progress. SAFe, designed for large-scale enterprises, offers moderate improvements across all metrics, with a strong focus on escaped defect performance (30%). Lean, emphasizing process efficiency, shows steady performance across all areas but leads in escaped defect performance (35%), ensuring minimal quality issues in final deliverables. This comparison highlights how each framework prioritizes different aspects of agility, depending on project needs.

Table 3.1 each methodology, highlighting efficiency, quality improvement, collaboration, and scalability

Methodology	Efficiency Gain	Quality Improvement	Collaboration Enhancement	Scalability
Scrum	High	High	High	Moderate
Kanban	Moderate	Moderate	High	Low
XP	Moderate	High	High	Low
SAFe	High	High	High	High
Lean	High	High	Moderate	High

IV CASE STUDIES

Case Study 1: Scrum - Spotify

Sprint Velocity: Spotify significantly improved its velocity after adopting Scrum. By breaking down development into small, manageable "sprints", Spotify teams delivered new features regularly and quickly adapted to user feedback, enhancing their competitive advantage.

Team Collaboration: Spotify's Squads, as Scrum-like Agile units, fostered strong team collaboration, where autonomy within squads led to more agile decision-making and reduced communication barriers between developers, designers, and product owners.

Customer Satisfaction: Continuous improvement through regular releases and client feedback loops resulted in better customer satisfaction. Features like personalized playlists have been developed iteratively, leading to high engagement and satisfaction rates [28].

Case Study 2: Kanban - Toyota

Workflow Efficiency: Toyota is a prime example of a Kanban system in action. Their production lines and software development teams benefit from the

Kanban approach by focusing on workflow visualization and eliminating bottlenecks. Real-time data shows continuous process optimization, reducing lead time and improving production efficiency[29].

Team Autonomy: Kanban has enabled Toyota software development teams to self-organize work. This autonomy has led to more responsive teams capable of optimizing workflows without centralized decision-making.

Continuous Improvement: Ongoing data from Toyota's software division indicates continual improvement, reducing waste and rework, enhancing software production speed.

Case Study 3: XP (Extreme Programming) - BMW

Code Quality: BMW's automotive tech division adopted XP practices for developing embedded software for its luxury vehicles. The emphasis on test-driven development (TDD), pair programming, and continuous integration dramatically improved the quality of their embedded systems, reducing the likelihood of defects in high-risk components.

Team Productivity: Initial team adjustment times were slow, but after full implementation, BMW's XP teams saw an increase in productivity due to improved collaboration and shared responsibility for quality.

Customer Satisfaction: BMW's XP-driven development helped bring in direct customer feedback, ensuring alignment with vehicle buyers' needs. Increased feedback loops, combined with fast feature iteration, boosted satisfaction with software capabilities in newer models. System is to deliver environmental data to remote locations[30]

Case Study 4: SAFe - Cisco

Alignment and Coordination: Cisco adopted the SAFe (Scaled Agile Framework) for improving coordination across its product development teams working on various aspects of networking solutions, like routers and switches. Teams working on hardware, software, and cloud services benefited from SAFe's structured methodology, enhancing overall alignment across technical departments.

Scalability: For large-scale projects across multiple teams, SAFe allowed Cisco to scale Agile practices successfully, providing a repeatable framework that could be extended globally across its multiple development and production sites.

Organizational Agility: By implementing SAFe, Cisco enhanced its ability to adapt rapidly to market trends in networking technology, staying competitive in a fast-moving tech industry and being able to pivot based on customer demands faster.

Case Study 5: Lean - Intel

Waste Reduction: Intel successfully implemented Lean principles to streamline its software and hardware development processes. By eliminating unnecessary tasks, optimizing workflows, and reducing redundancy in development cycles, Intel minimized delays in chip production and software deployment.

Faster Time-to-Market: Lean principles helped Intel accelerate product releases, particularly in the development of processors. By focusing on delivering "just-in-time" updates and reducing unnecessary work-in-progress, Intel shortened the cycle time for new chip designs, ensuring faster market entry for their latest technologies.

Continuous Improvement (Kaizen): Intel embraced continuous improvement by analyzing past development cycles and implementing feedback-driven enhancements in their Agile teams. Teams adopted Lean Startup principles when testing new ideas, rapidly iterating on software and hardware prototypes, ensuring a balance between innovation and efficiency.

Customer-Centric Approach: Intel improved customer satisfaction by aligning software and chip development with real-time market demands, integrating customer feedback into product iterations, and ensuring that innovations met actual needs rather than speculative advancements.

V CONCLUSION

This comparative analysis of Agile methodologies highlights their distinct strengths in software development. Kanban excels in lead time reduction and productivity, making it ideal for teams that require continuous workflow optimization. XP focuses on technical excellence, achieving significant defect rate reduction and cycle time improvements through practices like test-driven development. Scrum balances collaboration, velocity, and productivity, fostering structured teamwork and iterative delivery. SAFe, designed for large-scale enterprises, ensures high scalability

while maintaining steady performance across various metrics. Lean emphasizes efficiency and defect reduction, minimizing waste and improving overall process quality. Ultimately, selecting the right Agile framework depends on the organization's goals, team structure, and project complexity. By leveraging the strengths of each methodology, businesses can enhance efficiency, product quality, and overall team performance.

References

[1] M. Dülgerler, “Making better, more responsive organizations", In Proceedings of the PMI® Global Congress 2015 - EMEA, London, England, 2015.

[2] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile Software Development: Methodologies and Trend”, International Journal of Interactive Mobile Technologies, vol. 14, no. 11, pp. 246-270, 2020.

[3] S. Lee, and H. S. Yong, "Agile Software Development Framework in a Small Project Environment”, Journal of Information Processing Systems (JIPS), vol. 9, no. 1, pp. 69-88, 2013.

[4] T. Dingsøyr, and N. Moe, "Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda”, In Proceedings of the 15th International Conference on Agile Software Development, Rome, Italy, pp. 1-9, 2014.

[5] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review”, Journal of Systems and Software, vol. 119, pp. 87-108, 2016.

[6] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at Ericsson: a case study”, Empirical Software Engineering, vol. 23, no. 5, pp. 2550-2596, 2018.

[7] F. Almeida, E. Miranda, and J. Falcão, "Challenges and facilitators practices for knowledge management in large-scale scrum teams”, Journal of Information Technology Case and Application Research, vol. 21, no. 2, pp. 90-102, 2019.

[8] D. Marjanovic, M. Storga, S. Skec, N. Bojcetic, and N. Pavkovic, "Agile beyond software – a study of a large scale agile initiative”, In Proceedings of the Design 2018 15th International Design Conference, Dubrovnik, Croatia, pp. 2055-2062, 2018.

[9] Saleh S., Huq S., and Rahman M. (2019). Comparative study within Scrum, Kanban, XP focused on their practices. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-6).

[10] Sharma P, and Hasteer N. (2016). Analysis of linear sequential and extreme programming development methodology for a gaming application. In 2016 International Conference on Communication and Signal Processing (ICCSP) (pp. 1916-1920).

[11] Herdika H., and Budiardjo E. (2020). Variability and commonality requirement specification on agile software development: Scrum, XP, Lean, and Kanban. In 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE) (pp. 323-329) [16] Ibid (14) [17] Ibid (14) [18] Ibid (17)

[12] Wang X. (2011). The combination of agile and lean in software development: an experience report analysis. In 2011 Agile Conference (pp. 1-9). [20] Ibid (17)

[13] Doheny, M., Nagali, V., and Weig, F. (2012). Agile operations for volatile times. McKinsey Quarterly, 3, 126-131.

[14] Martins, A. M., Correia, P., & Gouveia, R. (2023). The impact of the Russia–Ukraine war on the world‟s largest listed insurance firms. The Geneva Papers on Risk and Insurance-Issues and Practice, 1-25

[15] Aimar, C., and Smith, D. K. (2021). VUCA: A management tool for dealing with challenges in changing environments. Global Journal of Entrepreneurship (GJE), 5

[16] JanaPócsová,Dagmar,Bednárová,Gabriela,Bogdanovská,AndreaMojžišová(2020)“Implementation of Agile Methodologies in an Engineering Course” 2020, 10(11), 333; https://doi.org/10.3390/educsci10110333, 17 November 2020

[17] Gleiston Guerrero-Ulloa,Carlos Rodríguez-Domínguez,Miguel J. Hornos (2023) “Agile Methodologies Applied to the Development of

Internet of Things (IoT)-Based Systems: A Review" 2023, 23(2), 790; https://doi.org/10.3390/s23020790,, 10 January 2023

[18] Patricia Losana,John W. Castro,XavierFerre,Elena Villalba-Mora,Silvia T. Acuña (2021) "A Systematic Mapping Study on Integration Proposals of the Personas Technique in Agile Methodologies" 2021, 21(18), 6298; https://doi.org/10.3390/s21186298,, 20 September 2021

[19] Kruchten, Philippe. 2010. "Software Architecture and Agile Software Development—A Clash of Two Cultures?" Paper presented at the ACM/IEEE 32nd International Conference on Software Engineering, Cape Town, South Africa, 2- 8 May

[20] Jason Leong,Kiu May Yee,OnalethataBaitsegi,LingesvaranPalanisamy,R. Kanesaraj Ramasamy (2023) "Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability" 2023, 15(2), 1121; https://doi.org/10.3390/su15021121, 6 January 2023\

[21] SanjayaChathuranga,ShanJayasinghe,JurgitaAntucheviciene,RuwanWickramarachchi,Nilan Udayanga,W. A. S. Weerakkody (2023) "Practices Driving the Adoption of Agile Project Management Methodologies in the Design Stage of Building Construction Projects" 2023, 13(4), 1079; https://doi.org/10.3390/buildings13041079, 19 April 2023

[22] Daniel Soares,Francisco José Gomes da Silva,Sandra Cristina Faria Ramos,KonstantinosKirytopoulos,José Carlos Sá,Luís Pinto Ferreira (2022) "Identifying Barriers in the Implementation of Agile Methodologies in Automotive Industry" 2022, 14(9), 5453; https://doi.org/10.3390/su14095453, 1 May 2022

[23] Fernando Almeida,Blaskovics Bálint(2024) "Approaches for Hybrid Scaling of Agile in the IT Industry: A Systematic Literature Review and Research Agenda" 2024, 15(10), 592; https://doi.org/10.3390/info15100592, 28 September 2024

[24] Peter Kokol (2022)"Agile Software Development in Healthcare: A Synthetic Scoping Review" 2022, 12(19), 9462; https://doi.org/10.3390/app12199462, 21 September 2022

[25] Manuel A. Quintana,Ramón R. Palacio,,Gilberto Borrego Soto,Samuel González-López (2022)"Agile Development Methodologies and Natural Language Processing: A Mapping Review" 2022, 11(12), 179; https://doi.org/10.3390/computers11120179, 7 December 2022

[26] RubabWafa,MuhammadQasim,Khan,FazalMalik,Akmalbek,BobomirzaevichAbdusalomov, YoungImCho,RomanOdarchenko (2022) "The Impact of Agile Methodology on Project Success, with a Moderating Role of Person's Job Fit in the IT Industry of Pakistan" 2022, 12(21), 10698; https://doi.org/10.3390/app122110698, 22 October 2022

[27] Sharad Kumar Jain ,Dr.Varsha Jotwani "Agile methodologies in project management: evolution, applications, and key challenges" INTERNATIONAL JOURNAL OF RECENT TECHNOLOGY SCIENCE& MANAGEMENT,2025; https://doie.org/10.0129/IJRTSM.2025542011 .

[28] Neeta Verma ,Dr.Varsha Jotwani "Viable Cloud Computing Setup Based On Service-Level Portfolio" Webology (ISSN: 1735-188X) Volume 18, Number 6, 2021.

[29] Bhavna Bhadkare, Dr. Varsha Jotwani "Enhancing Face Recognition Accuracy On Low-Resolution Databases Using Interpolation Techniques And Feature Extraction Techniques." Journal of Advanced Zoology ISSN: 0253-7214 Volume 45 Issue -3 Year 2024 Page 465-482

[30] Ashish Dibouliya, Dr. Varsha Jotwani "A review on: a hybrid smart IOT based real time environment monitoring system", European Chemical Bulletin, ISSN 2063-5346, Eur. Chem. Bull. 2023,12(Special Issue 5),11615-1624.