



Agile Methodologies in Software Development: Increasing Team Productivity

Santosh Panendra Bandaru*

Independent Researcher, USA.

*Corresponding Author

Santosh Panendra Bandaru

Independent Researcher,
USA.

Article History

Received: 28.04.2024

Accepted: 13.05.2024

Published: 26.06.2024

Abstract: Software development is now a crucial component of almost every aspect of business, thanks to the growth of the Internet. Businesses are under pressure to expand their product offerings to include web-based services due to customers' overwhelming need for ease and immediacy. In order to satisfy client demands, more and more resources are being devoted to the creation of lucrative software. Companies want to make as much money as possible, therefore they need to allocate these resources as efficiently as possible to keep expenses down. By putting into practice a process model that optimises their resources into high-quality goods, this may be accomplished. Agile software development encompasses a collection of iterative software development approaches in which cross-functional self-organising teams collaborate to evolve requirements and solutions. The experiences of individuals working on real-world projects using Agile software construction methodology following various variants in varying team sizes were investigated by visiting various software houses in a developing nation to ascertain their choice of using Agile software development the method in larger team sizes. Many software companies are switching from conventional techniques to Agile development methodologies as a result of the growing popularity of these approaches. Agile is people-focused and flexible rather than predictive. It promotes a small, cooperative team that collaborates closely. However, team size is a factor that is further limited by human characteristics. These crucial elements are often disregarded when putting Agile into practice. The purpose of this research is to determine the fundamental human elements that need be taken into account while implementing Agile in order for a team to function well. The approach is a study of three Agile teams of varying sizes using Scrum to produce products based on the same technology. Both subjective and objective metrics were used, and a poll backs up the findings. The findings unequivocally demonstrate that choosing the appropriate individuals for the right team is essential to the success of agile approaches.

Keywords: Agile Software, Agile Methodologies, Developing Products, Software Development, Self-Organizing Teams, Profitable Software, Cross-Functional, Subjective Measures.

Cite this article:

Bandaru, S. P., (2024). Agile Methodologies in Software Development: Increasing Team Productivity. *ISAR Journal of Science and Technology*, 2(6), 42-48.

I. Introduction

To guarantee that all members of the development team, including management, engineers, and customers, are aware of the workflow's expectations and to establish a schedule for software project deliverables, a process model is required [1]. Despite the waterfall model's long history, businesses are starting to recognise its limitations. Its linear process of developing requirements, putting the whole functionality into practice, and then testing makes it difficult to adjust to changing customer demands and puts the project at significant risk [1, 9]. The effect of adopting an agile approach on such risk and related expenses is examined in my study [2, 5].

Agile is an iterative development method in which features are added in increments of two to four weeks. As a result, during the software development cycle, all three of the primary components are replicated [8, 9]. Because a tiny bit of functionality is provided

every few weeks and can be readily changed, this accommodates changes in customer preferences. Because there is less risk than there would be under the waterfall approach, software projects using the agile framework would have lower capital costs [8, 9]. The end product is almost exactly in line with client expectations because to this process's high degree of adaptability and openness to ongoing input, which raises customer satisfaction and demand and boosts revenues. However, only a significant degree of agility, or an adequate use of an agile technique, may result in such lower software development costs [8, 9].

Many writers stress the need of maintaining a small team size while using any kind of agile development technique. The ideal team size to get the best development outcomes is a topic of disagreement among writers [8, 9]. According to earlier research, an agile team should include no more than 15 persons, but no more than 20. In bigger teams, the quality of decision-making diminishes because the team is confused by the choices, which are more

ambiguous [9, 19]. The team becomes more complacent since nobody begins working on the project until they get explicit instructions, and it may be difficult to keep track of who is responsible for what in a big team. Nevertheless, every previous study focusses mostly on other aspects of agile development, with a passing mention of team size [19, 25].

1.1 Background

Few efforts have been made to quantify and demonstrate how adopting an agile development process model affects overall software quality and costs. While Kai Petersen, for and Claes Wohlin found several benefits by switching from a plan-driven to an incremental software development approach by conducting a case study at Ericsson AB in Sweden [1, 8], Jaana-Nyford and Mira Kajko-Mattsson from Stockholm University in Sweden are currently integrating agile methods with risk management [25, 26].

- **Requirements:** Before any feature can be implemented, plan-driven, or classical, approaches need thorough and in-depth planning. Much time and effort are wasted outlining a comprehensive list of requirements because many documented and validated demands [1, 2] that were only compiled at the beginning of the development process have to be discarded and reworked during the implementation and testing phases [2, 3] due to changing customer needs. As a result, there seems to be waste since the percentage of executed requirements compared to all requirements specified is tiny.
- **Customer Needs:** Customers only used a tiny percentage of the generated application code in a research that included 400 projects created utilising a waterfall methodology [2, 4]. The absence of a chance to learn about changing client wants is confirmed by this research. It is also obvious that a plan-driven process model does not provide a channel for feedback or a chance to address any misconceptions about client needs [4, 5]. It is becoming harder and more costly to identify current customer demands and modify requirements and implementation in accordance with them as the three primary steps of software development—planning, implementation, and testing—are carried out in a linear, bulk approach [5, 6].
- **Number of Faults:** The large number of errors discovered during the testing stage is another key problem with plan-driven development. Testing is the first thing that is lost when previous stages of the process model take an unanticipatedly lengthy time since it is only done at the end of development [5, 6]. Test coverage is poor because not enough testing can be performed to identify enough flaws in time to fulfil a deadline.

Even if agile development has its share of problems, the quantity and severity of problems that affect a plan-driven approach's expenses and return on investment much exceed those of an agile methodology [5, 6]. High testing lead times, limited test coverage, and several teams needing a high level of communication and coordination from the project manager are common problems that agile still faces despite moving away from conventional approaches [6, 7].

- **Testing Lead Times:** A piece of functionality that is only realised and implemented too late in an iteration to be

tested, forcing it to wait until the following iteration, might result in high testing lead times [5, 6].

- **Test Coverage:** A substantial test environment with test cases for each incremental piece of functionality is necessary because agile methodologies need continuous testing (testing at the conclusion of each iteration) [6, 8]. Implementing functionality in tiny chunks while paying careful consideration to detail and current client demands is a requirement of agile development.
- **Communication & Coordination:** Management overhead is the last frequent problem with agile development [5, 9]. Strong cooperation is essential for the effective implementation of an agile approach, thus the project manager must stay engaged in team dynamics to encourage a feeling of belonging and a commitment to the calibre of the finished product [8, 9]. To guarantee that all staff members from all process phases—business analysts, software developers, testers, and others—are cooperating to create a successful product, this calls for extensive communication and coordination between team members and management [11].

According to the software engineering conceptual framework known as "agile project management," software is developed in a comparatively short amount of time and goes through several iterations before being released in a stable state. A collection of concepts that emphasise client value, iterative and incremental transportation, intensive communication, small integrated teams, self-organization, and tiny and continual improvements are the foundation of Agile, according to [12, 13]. Agile management is said to function best in small teams [14, 16]. The ideal Agile project management group is small, collaborative, [15], has daily in-person communication, and has a maximum of nine members. Furthermore, Agile approaches like Scrum suggest that a team should consist of seven members, plus or minus two [15, 16].

The purpose of this article is to examine an Agile project management approach, test it out on actual projects, and determine the human elements that must be taken into account for an Agile team to be successful. A few psychological elements that might influence teamwork are somewhat taken into account in this research, including working memory [16–17], ideal experience, and suggested team size based on social behaviours. Three example Agile teams of varying sizes will be utilised to do this, and they will all be run using the same Agile principles. Given that each team has all the required skill sets and comparable team member profiles, the findings will assist in identifying which team will function the best and provide the greatest outcomes [18, 19].

Agile suggests that teams should include all the necessary skill sets rather than just coders. It is often held that Agile is a one-size-fits-all idea that is thus scalable, even if many Agile approaches, like Scrum, do have a suggested team size [18, 19]. However, this raises the issue of how large an Agile team may go before losing effectiveness. Team size may directly affect group performance and communication, according to study findings [19, 20].

While software project management is a more technical field, agile management is more people-focused [20, 21]. When switching from conventional to Agile methods, project managers often find that they are spending more time overseeing projects than overseeing the team members. This suggests that Agile teams may

not be operating at their best since human elements are often overlooked while managing them [22].

II. Related Work

This paper examines previous studies and research on Agile development teams.

2.1 Agile and People Management

Several models, like Tuckman's model and "The Five Dysfunctions," serve as the foundation for people and team management. Each of these models calls for a high level of engagement among team members. More contact is needed and team management becomes more challenging as the number of members increases [11]. To facilitate efficient collaboration and project management in an Agile team, the work management must specify the connections between the various responsibilities.

(H. Oktaba, 2017) [11] Agile ideas have been embraced by several software development organisations in an effort to boost productivity. Although Agile Software Development (ASD) promotes regular delivery of functional software, productivity in that setting is influenced by a variety of variables. In order to determine the factors that influence productivity, the most popular research techniques, the most researched rates of productivity, the principles associated with productivity, the most popular methods for agile development, and the application of productivity measurements in ASD, this paper will analyse the literature on productivity in ASD.

Rosa, P. F. (2011) [12] For an organisation to achieve the necessary levels of productivity and quality, software development methods are crucial. In order to capitalise on the advantages of both techniques, industry and academic colleagues have shown interest in the uncharted and understudied field of productivity analysis of agility and conventional development processes. The goal of this study is to define a hybrid process that benefits from both agile and conventional integration methods in order to explore viable possibilities for both. In a Brazilian CMMI v.1, empirical research was conducted with the goal of assessing the productivity effect of the suggested hybrid approach. 1 Medium-sized business, Maturity Level 2.

D. I. Sjøberg (2016) [13] Agile development relies heavily on small, self-directed teams. This article examines how collaboration quality affects agile software teams' performance, learning, and job satisfaction and if this impact is different from that of conventional software teams. Structural equation modelling was used to analyse a survey that was given to 477 respondents from 71 agile software development organisations across 26 firms. When collaborators and team leaders evaluated team performance, they discovered that collaboration quality had a favourable impact on team performance.

(A. Uldahl, 2014) [14] With an empirical investigation in the form of a case study that shows the beneficial effects of the agile development and project management approach Scrum on information systems and software development productivity, this research adds to the body of knowledge in the field of information technology development (ISD). It also offers a helpful operationalisation of the concept through seven productivity indicators that have been identified.

(B. Kola, 2014) [15] The primary driving force behind this study is the high failure rate of projects related to software development, which directly affects the economic value of the organisations where they are implemented by lowering revenues and lowering staff productivity. The aim of this research is to determine the best methods for measuring and enhancing team productivity. It also seeks to determine the degree to which agile methods, lean principles, and the team's "soft skills" can be integrated to produce high-quality outcomes. Method: To find out how much the opinions of the academics and researchers whose work is used to construct the theoretical structure of this dissertation apply to actual organisations operating in the software development industry, a qualitative study was conducted. It involved surveys with only Scrum masters and online surveys with both Scrum masters and practitioners.

In 2022, Lindsjorn, Y. [16] In software development, teamwork is essential, especially in cross-functional agile development teams where members collaborate closely to create a seamless software solution. Working in a team is difficult; previous research shows difficulties with leadership, communication, learning, and setting priorities. Agile approaches, practitioner literature, general collaboration studies, and an expanding corpus of empirical research on teamwork in the particular context of agile development of software, however, provide a wealth of guidance for teams.

Several soft variables may influence the effectiveness of team members in a software development team, according to 2011 research by Sudhakar et al. A development team's success is said to be influenced by four types of factors: organisational, environmental, non-technical, and technical [15, 16]. The following soft elements were suggested by the authors as potential influences on software team performance:

- **Team climate:** Same views and aspirations to accomplish organisational objectives.
- **Team diversity:** For example, differences in team member credentials, gender, ethnicity, experience levels, and talents.
- **Team innovation:** Innovative methods of problem-solving and abilities that provide value.
- **Team member competencies and characteristics:** Team members' technical and interpersonal skills that affect familiarity and cooperation [18].
- **Team leader behaviour:** Increased personnel manager and a facilitator role, less micromanagement.
- **Top management support:** Management's dedication to the project [19].

III. Proposed Methodology

The process for determining the human elements that might impact the effectiveness of an Agile team is the subject of the next chapter [19, 20].

3.1 Company background

The design and development teams of a company that specialises in goods and transportation services participated in this investigation [29, 30]. With extensive operations in the Middle East, Asia, South America, and Australia, the group is a significant

force in Europe. The organization's IT department has been in operation for more than 30 years and has matured to a respectable level in the software sector [21, 22]. Prior to switching to Agile Methodology with Scrum, it used a number of development approaches, including a completely tailored Rational Unified Process.

3.2 Study of Agile Teams

Over the course of six months, this research was conducted utilising Agile Methodology on three projects of varying sizes. Not all of the projects have been finished in the entire six months; some have been finished sooner Table 1 [2, 23].

Table 1 Project Size Overview.

Project	Initial Estimate-MD
Project A	1800 (89.6-man yrs.)
Project B	800 (3.7-man yrs.)
Project C	100 (0.9-man yrs.)

The technologies used and the degree of complexity for each project that has been assessed in accordance with the organization's requirements are listed in Table 2 below [24, 26].

Table 2 An Overview of Complexity and Technologies. [25]

Project	Language/Technology	Complexity
Project A	Oracle, WebSphere Portal, Java Services, Spring MVC, and Free-Marker.	High
Project B	Oracle Database, WebSphere Portal, Spring MVC, and Java Services.	High
Project C	Oracle Database, WebSphere Portal, Java Services, Spring MVC, and Free Marker.	High

A project manager owns the whole project, whereas a project leader leads each team [25, 26]. An overview of the different duties assigned to each team is provided in Table 3 below:

Table 3 Project Team Overview. [26, 27]

Project	Business Analysis	Analysis Programmer	Text Analysis	Technical Architect	Total
Project A	2	11	3	0	22
Project B	1	7	4	1	12
Project C	1	2	6	2	6

$$Rework\ Level = \frac{Time\ Spent\ fixing\ defect}{Time\ spent\ on\ the\ development} \times 100 \dots\dots\dots 1$$

IV. Results and Discussion

It should be mentioned that the findings of this study should not be interpreted as a general rule or a standard in the software sector [28]. They do, however, provide light on the aspects that must be taken into account and the ideal choice for the size of an Agile team [28].

Following the conclusion of each project, the budget was compiled using JIRA and Clarity [28, 29]. The outcomes are as follows:

Project	Initial estimate (MD)	Budget speed (MD)	% Deviation
Project A	1800 (89.6-man yrs.)	3200 (16.6-man yrs.)	65
Project B	800 (3.6-man yrs.)	950 (4.9-man yrs.)	22
Project C	120 (0.9-man yrs.)	130 (0.9-man yrs.)	4

Table 4 Budget.

The following metrics were used to gauge how often releases were completed on schedule:

Table 5 On Time delivery. [29, 30]

Project	Releases Performance	Releasees on Time	% On Time
Project A	40	13	30
Project B	22	15	60
Project C	12	9	90

Each project's rework level has been determined as follows:

Table 6 Rework Level. [30]

Project	Actual Spend -MD	Rework-MD	Rework Level-%
Project A	3200 (26.9-man yrs.)	120	26
Project B	950 (2.96-man yrs.)	190	19
Project C	130 (5.96-man yrs.)	25	18

The following defect rates have been noted:

Table 7 Defect Rate.

Project	Actual Spend -MD	Rework-MD	Rework Level-%
Project A	3200 (19.6-man yrs.)	1960	1.9
Project B	950 (2.9-man yrs.)	396	3.9
Project C	130 (0.9-man yrs.)	49	2.5

The intricacy of communication was one of the main obstacles that the Agile team participants had to overcome. Table 8. Effective communication was not a problem for the team members involved in Project C [28, 29]. However, this was much more challenging for teams A and B.

Table 8 Communication Channels.

Project	No. of People on team	No of communication Channels
Project A	23	236
Project B	19	69
Project C	5	10

V. Conclusion

This research was conducted inside an enterprise using Scrum and was solely based on the Agile Methodology. No initiatives using other kinds of development approaches were taken into account. Therefore, it is essential that any organisation determine if its business processes, projects, operational structure, and corporate culture are appropriate for the implementation of an Agile Project Management Methodology before deciding to adopt one. The purpose of this research was to assess the human elements that may influence an Agile team's success. A limited collection of related projects that varied only in team size and size were used for the investigation. Team leaders will get important knowledge from this activity that will help them choose the right kind and quantity of personnel for each project. Software companies, who will be in a better position to determine if agile may work for them, would also greatly benefit from it. Future research will be necessary to determine if initiatives with different organisation, complexity, and technological capabilities can achieve the same outcome.

VI. References

- [1] Nyfjord, J., & Kajko-Mattsson, M. (2008, May). Degree of agility in pre-implementation process phases. In *International Conference on Software Process* (pp. 234-245). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [2] Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study. *Empirical Software Engineering*, 15, 654-693.
- [3] Petersen, K. (2010, July). An empirical study of lead-times in incremental and agile software development. In *International Conference on Software Process* (pp. 345-356). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [4] Kruchten, P. (2011, May). A plea for lean software process models. In *Proceedings of the 2011 International Conference on Software and Systems Process* (pp. 235-236).
- [5] Rönkkö, M., Järvi, A., & Mäkelä, M. M. (2008). Measuring and comparing the adoption of software process practices in the software product industry. In *Making Globally Distributed Software Development a Success Story: International Conference on Software Process, ICSP 2008 Leipzig, Germany, May 10-11, 2008 Proceedings* (pp. 407-419). Springer Berlin Heidelberg.
- [6] Altameem, E. A. (2015). Impact of agile methodology on software development. *Computer and Information Science*, 8(2), 9.
- [7] Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). People factors in agile software development and project management. *International Journal of Software Engineering & Applications*, 3(1), 117.
- [8] Hamid, M. (2015). Agile Testing. *International Journal of Scientific and Research Publications*, 5(9).
- [9] Rajasekhar, P., & Shafi, R. M. (2014). Agile software development and testing: approach and challenges in advanced distributed systems. *Global Journal of Computer Science and Technology*, 14(1), 7-10.
- [10] Lindsjörn, Y., Sjøberg, D. I., Dingsøyr, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in

- software development: A survey of agile development teams. *Journal of systems and software*, 122, 274-286.
- [11] Ramírez-Mora, S. L., & Oktaba, H. (2017, October). Productivity in agile software development: a systematic mapping study. In *2017 5th international conference in software engineering research and innovation (CONISOFT)* (pp. 44-53). IEEE.
- [12] de Souza Carvalho, W. C., Rosa, P. F., dos Santos Soares, M., da Cunha Jr, M. A. T., & Buiatte, L. C. (2011, November). A comparative analysis of the agile and traditional software development processes productivity. In *2011 30th International Conference of the Chilean Computer Science Society* (pp. 74-82). IEEE.
- [13] Lindsjörn, Y., Sjøberg, D. I., Dingsøyr, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of systems and software*, 122, 274-286.
- [14] Kautz, K., Johanson, T. H., & Uldahl, A. (2014). The perceived impact of the agile development and project management method scrum on information systems and software development productivity. *Australasian Journal of Information Systems*, 18(3).
- [15] Kola, B. (2014). Thinking Lean in Agile Software Development Projects. A qualitative study to Identify ways to improve productivity and increase business value.
- [16] Strode, D., Dingsøyr, T., & Lindsjorn, Y. (2022). A teamwork effectiveness model for agile software development. *Empirical Software Engineering*, 27(2), 56.
- [17] Bahli, B., & Buyukkurt, M. D. (2005). Group performance in information systems project groups: An empirical study. *Journal of Information Technology Education: Research*, 4(1), 97-113.
- [18] Bustamante, A., & Sawhney, R. (2011). Agile XXL: Scaling Agile for Project Teams, Seapine Software.
- [19] Brady, K. (2006). AGILE/SCRUM Fails to get to grips with Human Psychology.
- [20] Conboy, K. Coyle, S. Lero, X.W. Pikkarainen, M., (2011). People over Process: Key Challenges in Agile Development July/August 2011 (vol. 28 no. 4) pp. 48-57
- [21] Csikszentmihalyi, M., & Csikszentmihaly, M. (1990). *Flow: The psychology of optimal experience* (Vol. 1990, p. 1). New York: Harper & Row.
- [22] Fayol, H. (Ed.). (1918). *Administration industrielle et générale: prévoyance, organisation, commandement, coordination, contrôle*. H. Dunod et e. Pinat.
- [23] Hastie, S., & Engineer, C. K. (2004, September). The Agile Mindset: what does it take to make this stuff work?. In *Software Education Associates Ltd, Agile Development Conference Wellington & Sydney*.
- [24] Huckman, R. S., Staats, B. R., & Upton, D. M. (2009). Team familiarity, role experience, and performance: Evidence from Indian software services. *Management science*, 55(1), 85-100.
- [25] Johnson, R. (2008). Six Principles of Effective Management. Retrieved June, 23, 2017.
- [26] Kemerer, C. F. (1989, January). An agenda for research in the managerial evaluation of computer-aided software engineering (CASE) tool impacts. In [1989] *Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences. Volume II: Software Track* (Vol. 2, pp. 219-228). IEEE.
- [27] Lencioni, P. (2002). The five dysfunctions of a team. *San Francisco: Jossey-Bass*. hwy.
- [28] Ong, A., Tan, G. W., & Kankanhalli, A. (2005, July). Team expertise and performance in information systems development projects. In *Proceedings of the 9th Asia Pacific Conference on Information Systems, Bangkok, Thailand, July* (pp. 7-10).
- [29] Standish Group., (2009). Chaos Report 2009.
- [30] Purna Sudhakar, G., Farooq, A., & Patnaik, S. (2011). Soft factors affecting the performance of software development teams. *Team Performance Management: An International Journal*, 17(3/4), 187-205.
- [31] Sakariya, A. B. (2024, December). DIGITAL TRANSFORMATION IN RUBBER PRODUCT MARKETING. In *International Journal for Research Publication and Seminar* (Vol. 15, No. 4, pp. 118-122). <https://doi.org/10.36676/jrps.v15.i4.18>
- [32] Ashish Babubhai Sakariya. (2024). Sustainable Marketing Approaches for the Rubber Industry. *International Journal of Research and Review Techniques*, 1(1), 43–50. Retrieved from <https://ijrrt.com/index.php/ijrrt/article/view/218>
- [33] Gadhiya, Y., Gangani, C. M., Sakariya, A. B., & Bhavandla, L. K. (2024). Emerging Trends in Sales Automation and Software Development for Global Enterprises. *International IT Journal of Research*, ISSN: 3007-6706, 2(4), 200-214. <https://itjournal.org/index.php/itjournal/article/view/86>
- [34] Ashish Babubhai Sakariya. (2023). The Evolution of Marketing in the Rubber Industry: A Global Perspective. *International Journal of Multidisciplinary Innovation and Research Methodology*, ISSN: 2960-2068, 2(4), 92–100. Retrieved from <https://ijmirm.com/index.php/ijmirm/article/view/175>
- [35] Ashish Babubhai Sakariya, " Leveraging CRM Tools to Boost Marketing Efficiency in the Rubber Industry , International Journal of Scientific Research in Science, Engineering and Technology(IJSRSET), Print ISSN : 2395-1990, Online ISSN : 2394-4099, Volume 4, Issue 6, pp.375-384, January-February-2018.
- [36] Ashish Babubhai Sakariya, " Impact of Technological Innovation on Rubber Sales Strategies in India , International Journal of Scientific Research in Science, Engineering and Technology(IJSRSET), Print ISSN : 2395-1990, Online ISSN : 2394-4099, Volume 6, Issue 5, pp.344-351, September-October-2019.
- [37] Gangani, C. M. (2024). AI in Insurance: Enhancing Fraud Detection and Risk Assessment. *International IT Journal of Research*, ISSN: 3007-6706, 2(4), 226-236. <https://itjournal.org/index.php/itjournal/article/view/91>
- [38] Gangani, C. M. (2024). Automated Data Integrity Checks for Financial Software Systems. *Journal of Sustainable Solutions*, 1(4), 197–207. <https://doi.org/10.36676/j.sust.sol.v1.i4.52>

- [39] Gangani, C. M. (2019). "Applications of Java in Real-Time Data Processing for Healthcare", International Journal of Scientific Research in Science, Engineering and Technology(IJSRSET), Print ISSN : 2395-1990, Online ISSN : 2394-4099, Volume 6, Issue 5, pp.359-370,
- [40] Gangani, C. M. (2020). "Data Privacy Challenges in Cloud Solutions for IT and Healthcare", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 7 Issue 4, pp. 460-469, July-August 2020. Journal URL : <https://ijsrst.com/IJSRST2293194> | BibTeX | RIS | CSV
- [41] Bhavandla, L. K. (2024). Cloud Compliance Systems: Trends and Future Directions. *International IT Journal of Research*, ISSN: 3007-6706, 2(4), 215-225. <https://itjournal.org/index.php/itjournal/article/view/87>
- [42] Bhavandla, L. K. (2023). International Journal of Computer Science and Mobile Computing, Vol.12 Issue.10, pg. 89-100.
- [43] Bhavandla, L. K. (2024). Using AI for Real-Time Cloud-Based System Monitoring. *Journal of Sustainable Solutions*, 1(4), 187–196. <https://doi.org/10.36676/j.sust.sol.v1.i4.51>
- [44] Gadhiya, Y. (2024). AI-Based Automation for Employee Screening and Drug Testing. *International IT Journal of Research*, ISSN: 3007-6706, 2(4), 185-199. <https://itjournal.org/index.php/itjournal/article/view/85>
- [45] Yogesh Gadhiya. (2022). Designing Cross-Platform Software for Seamless Drug and Alcohol Compliance Reporting. *International Journal of Research Radicals in Multidisciplinary Fields*, ISSN: 2960-043X, 1(1), 116–126. Retrieved from <https://www.researchradicals.com/index.php/rr/article/view/167>