

Implementing Agile Practice To Improve Team Productivity

Odofo, Ebitomo Nina¹ and Professor Edwinah Amah²

¹Department Management, Faculty Of Management Science, University Of Port Harcourt, Nigeria

ebitomoodofori@gmail.com

²Department of Management, Faculty of Management Science, University of Port Harcourt, Nigeria

Abstract: Agile development methodologies have transformed the landscape of software engineering, emphasizing iterative progress, collaboration, and flexibility. This study explores the implementation of agile practices to improve team productivity and to also briefly look at the impact of agile development strategies on team productivity in order to further buttress my point - focusing on how these approaches affect productivity, performance, efficiency, and overall output in software development teams. Agile methodologies, including Scrum, Kanban, and Extreme Programming (XP), have become popular due to their iterative nature and ability to adapt to changing requirements. By facilitating frequent feedback and continuous improvement, these strategies aim to improve team productivity and project outcomes. This study gave a comprehensive analogy for right conditions for agile mythology, stating in a tabular form, conditions: favourable and unfavourable, for a better implementation of agile practice. One of the key findings is that agile development enhances productivity by promoting a culture of continuous improvement and regular feedback. The flexibility inherent in agile methods allows teams to respond swiftly to changing requirements and unexpected challenges, contributing to improved project delivery and stakeholder satisfaction. However, the study also identifies challenges associated with agile adoption. Teams face difficulties in maintaining consistency in agile practices, especially in organizations with entrenched traditional workflows. Additionally, suggestions and recommendations were given for best agile practice most especially in the 21st century for continuous and effective improvement of employees' performance and team productivity.

Keywords • Agile Development • Team Productivity • Scrum • Kanban • Extreme Programming (XP) • Iterative Progress • Collaboration • Cycle Time • Defect Rates • Team Satisfaction • Continuous Improvement • Feedback • Agile Ceremonies • Flexibility • Project Management

INTRODUCTION

Agile methodology is a flexible and iterative approach to project management that emphasizes collaboration, adaptability, and continuous improvement (Junker *et al.*, 2022). It was initially developed for software development, but has since been applied to various industries and projects. Agile methodologies prioritize customer satisfaction, delivering working products quickly and frequently, and responding to change over following a rigid plan. The approach values individuals and interactions over processes and tools and encourages face-to-face communication and teamwork (Hemon *et al.*, 2020).

In Agile projects, the project is usually divided into several smaller cycles called sprints (Khan *et al.*, 2021). Each sprint is a mini-project within the predefined scope of work with a backlog and design, implementation, testing, and deployment stages (Malik *et al.*, 2021). The goal of each sprint is to deliver a potentially shippable increment of the product. The product gains new features after each iteration, leading to gradual project expansion. The risk of providing a product that may fail is reduced by validating the features in the earlier stages of development.

Agile methodologies have become increasingly popular as businesses strive to stay competitive in dynamic markets (Rigby *et al.*, 2018). By implementing Agile practices, teams can quickly adapt to changing requirements and customer needs, resulting in products that meet or exceed expectations. Agile project management has gained widespread adoption and is now considered the industry standard, with approximately 95 percent of organizations utilizing some form of Agile (Nguyen, 2024).

DISCUSSIONS

➤ PILLARS OF AGILE PROJECT MANAGEMENT

According to the Agile Manifesto, Agile project management has four main pillars. In agile project management, the following represents the pillars upon every other thing is built.

Individuals over processes and tools: One of the principles of Agile methodology asserts that face-to-face communication is the most efficient way to interact with the project team. This principle and encouragement to break down project silos are a recipe for collaborative teamwork. Despite advancements in technology and more remote-friendly work policies, the importance of face-to-face interaction remains unchanged. Agile teams prioritize collaboration and teamwork, placing less emphasis on independent work and rigid adherence to protocols (Paredes, 2024).

Working software over comprehensive documentation : The main focus for Agile teams is to develop functional software or products, while documentation is less of a priority (Weichbroth, 2022). The above assertion might seem to apply directly to software development teams. However, it has serious ramifications for project teams in various industries. It implies a result-oriented approach that measures progress based on achievement of stipulated goals. That does not mean that other aspects of the project are not important. It only means that small progresses should not shift the focus of the team away from the final goal. This disposition helps to galvanize efforts towards the overall goal behind initiation of agile teams, rather than being satisfied with success of individual teams (Žužek et al., 2020).

Customer collaboration over contract negotiation : Agile methodologies prioritize customer needs and center the development process around delivering value (Waldron, 2017). By comprehending and fulfilling the customer's requirements, the development team can create a product that meets their needs and achieves the desired outcomes. Agile teams utilize a process of continuous feedback and iteration, seeking input from customers throughout the development of a project to ensure the final product meets their expectations. Customers are given a significant role in guiding the direction of software development. As such, customer collaboration holds greater importance than the specifics of contract negotiation.

Adapting to changes rather than following a plan : Agile offers flexibility, enabling teams to promptly adjust their strategies and workflows without disrupting the whole project. Agile processes allow for quick strategy shifts without disrupting the project flow. Unlike other methods, such as the waterfall method, where each phase flows into the other, making strategy shifts challenging and disrupting the project roadmap. That's why Agile project management is favored in projects such as software development (Rigby & Hirotaka, 2016).

➤ CONDITIONS FOR IMPLEMENTATION OF AGILE METHODOLOGIES

Agile is not a panacea. It is most effective and easiest to implement under conditions similar to those commonly found in software innovation. The problem to be solved is usually complex; solutions are initially unknown, and product requirements will most likely change; the work can be modularized; close collaboration with end users (and rapid feedback from them) is feasible; and creative teams will typically outperform command-and-control groups (Rigby *et al.*, 2018).

These conditions exist for many product development functions, marketing projects, strategic-planning activities, supply-chain challenges, and resource allocation decisions. They are less common in routine operations such as plant maintenance, purchasing, sales calls, and accounting. And because agile requires training, behavioral change, and often new information technologies, executives must decide whether the anticipated payoffs will justify the effort and expense of a transition.

Table 1 Right conditions for Agile methodologies (Rigby *et al.*, 2018)

Conditions	Favourable	Unfavourable
Market environment	Customer preferences and solution options change frequently.	Market conditions are stable and predictable.
Customer involvement	(a) Close collaboration and rapid feedback are feasible. (b) Customers know better what they want as the process progresses.	(a) Requirements are clear at the outset and will remain stable. (b) Customers are unavailable for constant collaboration.
Innovation type	(a) Problems are complex, solutions are unknown, and the scope isn't clearly defined. (b) Product specifications may change. (c) Creative breakthroughs and time to market are important. (d) Cross-functional collaboration is vital.	(a) Similar work has been done before, and innovators believe the solutions are clear. (b) Detailed specifications and work plans can be forecast with confidence and should be adhered to. (c) Problems can be solved sequentially in functional silos.
Modularity of work	(a) Incremental developments have value, and customers can use them. (b) Work can be broken into parts and conducted in rapid, iterative cycles. (c) Late changes are manageable.	(a) Customers cannot start testing parts of the product until everything is complete. (b) Late changes are expensive or impossible.
Impact of interim mistakes	They provide valuable learning.	They may be catastrophic.

➤ AGILE METHODOLOGIES

The Agile framework encompasses various iterations. Here are some of the widely used Agile methodologies (Shirokova *et al.*, 2020):

Kanban: Kanban is a methodology that allows organizations to visualize their workflow and set limits for work in progress. This method is utilized when tasks are received unexpectedly and require immediate execution for other pending tasks without delay.

Lean: The lean methodology uses tools and principles to reduce waste to enhance process development speed. The goal is to optimize value and minimize waste. Lean is utilized in various industries that generate waste.

Scrum: Scrum is a framework utilized by teams to create a hypothesis, experiment, analyze the results, and make necessary modifications. Teams using this methodology can incorporate practices from various frameworks as required. The process involves cross-functional teams working on product development, with the work being divided into multiple 2-4 week iterations.

Crystal: This methodology focuses on human interactions rather than on tools and processes. It operates based on the principle that projects are dynamic and unique, intending to optimize processes. Crystal emphasizes team communication, continuous integration, user involvement, and adaptable processes.

Extreme Programming (XP): Extreme Programming (XP) is commonly utilized in software development. It emphasizes values that promote teamwork and includes regular releases and iterations but differs in its technical approach. XP is ideal for teams that need to swiftly release and address customer needs, focusing on the method of achieving this goal.

Adaptive Project Framework (APF): Also known as adaptive project management (APM). It was developed to address the potential for unforeseen factors during a project, particularly in IT, where traditional project management may be less effective. The framework operates under the premise that project resources are subject to change, including budget, timelines, and team member transitions. This approach prioritizes a project's existing resources over those that may be required.

Extreme Project Management (XPM): Projects with notable unpredictability often require this project management approach. It involves continuously adjusting processes until the intended outcome is achieved. This type of project includes numerous impromptu modifications, and it is not uncommon for teams to alter strategies weekly.

Adaptive Software Development (ASD): This methodology facilitates teams in adapting to changing requirements efficiently. The process emphasizes continuous adaptation, and the project phases of speculation, collaboration, and learning ensure continuous learning throughout the project.

Teams utilizing ASD may experience all three phases simultaneously due to the non-linear structure, resulting in overlapping phases. The fluidity of this management style may increase the probability of team members rapidly identifying and solving problems through constant repetition of the three phases, as opposed to traditional project management approaches.

Dynamic Systems Development Method (DSDM): This methodology emphasizes a comprehensive project lifecycle, resulting in a more rigid structure and foundation than other Agile methodologies. DSDM consists of four phases: feasibility and business study, functional mode or prototype iteration, design and build iteration, and implementation

Feature Driven Development (FDD): Customer input is vital to this methodology because the team prioritizes the features that customers need. This model allows teams to update projects frequently and quickly implement fixes if there are errors. The phases of the framework are constantly moving, making it easy to cycle through and update projects.

➤ AGILE METHODOLOGY BEST PRACTICES

Agile best practices are a set of guidelines and techniques that are commonly used by Agile teams to achieve maximum efficiency and productivity in their delivery process. These practices include regular meetings, self-organizing teams, and adopting a flexible and adaptive approach to change. Overall, Agile best practices aim to create a culture of continuous improvement and feedback, which helps teams deliver high-quality value quickly and efficiently. However, agile best practices also depend on the particular agile methodology in practice at a given organization (Masood *et al.*, 2018).

Agile Best Practices for Kanban Project Management

Kanban is a popular lean workflow management method for defining, managing, and improving services that deliver knowledge work. It helps to visualize work, maximize efficiency, and improve continuously. Work is usually represented on Kanban boards, allowing employees to optimize work delivery across multiple teams and handle even the most complex projects in a single environment. Originating from manufacturing, it later became a territory claimed by Agile software development teams. Recently, it started getting recognized by business units across various industries (Moloto *et al.*, 2020).

To ensure the successful implementation of Kanban in an Agile environment, it is recommended to adopt the following Kanban practices:

- **Visualizing the workflow:** Creating a visual representation of the workflow helps to identify bottlenecks, visualize the flow of work, and make the work more transparent.
- **Limiting work in progress:** Limiting the amount of work in progress helps to prevent multitasking and improve focus on completing one task at a time, thereby improving efficiency and reducing lead time.
- **Managing flow:** Kanban aims to help optimize flow, which can be achieved by monitoring flow metrics, identifying and addressing bottlenecks, and continuously improving the workflow.
- **Making process policies explicit:** Defining and communicating process policies clearly helps to ensure that everyone understands how work is supposed to be done, which reduces misunderstandings and promotes consistency.
- **Implementing feedback loops:** Kanban emphasizes the importance of getting feedback from customers, stakeholders, and team members to identify improvement areas.
- **Improving collaboratively:** Kanban is a continuous improvement process that encourages collaboration and experimentation to identify and solve problems, improve continuously, and evolve processes to meet the needs of customers better.

Agile Best Practices for Scrum Project Management

Scrum is often regarded as one of the most popular Agile frameworks. Statistical data indicate that Scrum is still the leading Agile approach employed by 87% of organizations, followed by Kanban and Scrumban (Rigby *et al.*, 2018). Here are some of the recommended Agile best practices for implementing Scrum within an Agile environment:

- **Foster a collaborative environment.** A recommended approach towards effective Scrum implementation is to foster a collaborative environment where the product backlog and product vision are co-created by the development team and stakeholders. This ensures a shared understanding and alignment between the two parties, thereby enhancing the chances of project success.
- **Hold daily stand-ups.** Stand-up meetings are short meetings of team members to discuss project progress. These meetings are designed to be kept to a maximum of 15 minutes to ensure they are efficient and productive. Incorporating stand-up meetings into product or project management is an effective way to monitor progress and keep all team members informed of updates.
- **Use a daily burndown chart to track the progress of sprints.** The burndown charts are a graphical representation of the work that has been accomplished in relation to the total work remaining against a given time frame.
- **Set communication guidelines for teams.** Developing a communication strategy that includes essential guidelines for teams can ensure smooth and effective communication. This practice can be particularly valuable for remote teams as it promotes transparency in reaching team goals.

Agile Best Practices for Extreme Programming (XP)

Extreme Programming (XP) is a software development framework that emphasizes teamwork, communication, simplicity, and customer satisfaction (Paredes, 2024). It is an Agile development approach involving continuous testing, integration, and delivery to ensure high-quality software that meets customer requirements. XP values include courage, communication, feedback, respect, and simplicity. In addition, it emphasizes practices such as pair programming, test-driven development, and incremental design. Some of the Agile best practices associated with Extreme Programming (XP) are:

- **Pair Programming:** Two developers can work together on the same code, taking turns between writing and reviewing code.
- **Test-Driven Development (TDD):** Write automated tests before writing the code.
- **Continuous Integration (CI):** Merge all code changes into a central repository multiple times a day and run automated tests to detect and fix issues early.
- **Refactoring:** Continuously improving the codebase by making small changes to improve its quality and maintainability.
- **Simple Design:** Keep the design of the software as simple as possible to avoid unnecessary complexity.
- **On-Site Customer:** Having a customer representative available to provide feedback and answer questions during development.

Agile Best Practices for Lean Development Model

The fundamental concept of lean project management involves the application of lean manufacturing principles to various project management processes (Parker *et al.*, 2015). These principles prioritize eliminating any activities or processes that don't contribute to the project's overall value. There are several Agile best practices associated with the lean development model, including:

- **Continuous Improvement:** Continuous improvement involves constantly reviewing and improving processes to increase efficiency and eliminate waste.
- **Visual Management:** This involves using visual aids such as Kanban boards, flowcharts, and other tools to help teams visualize work and identify bottlenecks.
- **Customer Collaboration:** Lean development emphasizes working closely with customers to understand their needs and deliver value quickly.
- **Small Batches:** To reduce waste and improve efficiency, work is done in small batches, with frequent feedback and adjustments.
- **Cross-functional Teams:** Lean development encourages cross-functional teams that include members with diverse skills and backgrounds to improve collaboration and problem-solving.

➤ AGILE BEST PRACTICES FOR PLANNING

Sometimes, it is said that there is no planning in Agile. This statement is far away from the truth. Below are some of the primary Agile practices for planning projects.

(a) **Plan on Multiple Levels and Prioritize Customer Collaboration:** One of the central Agile practices is to plan on multiple organizational levels going from the strategic initiatives all the way down to the individual tasks. As opposed to having one big planning phase, Agile teams spread it alongside the project's course and plan only for the short-term. Once a deliverable is released to the customer, the planning process can start again. Furthermore, Agile prioritizes customer collaboration above everything else, because this is what dictates the plan. In other words, Agile teams design their plans to focus on outcomes that produce customer value instead of outputs (Yu *et al.*, 2014).

To do this in practice, the Agile approach to project management emphasizes regular customer synchronization points in the process. Those are extremely important in an uncertain environment, because they enable fast feedback and feed valuable information to the next portion of the project plan. This practice of planning on multiple levels and prioritizing customer collaboration creates a highly adaptive planning process. Whenever there is a change in requirements, every level sends information to the one above it. This allows both teams and managers to apply small tweaks to their plans instead of large reworks, which lead to project failures.

(b) **Cascade Power Downwards:** For the Agile planning process to function correctly, management in organizations needs to adopt cascading power downwards. This means that team members should be encouraged to create their own plans based on some coarse-grained details only. In other words, managers in Agile are responsible for setting a product or service vision, making sure it's well-understood, and engaging in high-level planning. However, when it comes to developing a fine-grained path to achieving that vision, this is where individual team members become actively involved in the process as opposed to traditional project management (Ersoy & Mahdy, 2015).

This approach has several benefits over the conventional one. It involves the actual subject matter experts (team members), which increases their sense of belonging and produces more accurate plans. Furthermore, it enables responsiveness to changes as the high-level plans are only directions of how things should look instead of prescriptive documents (Serrador & Pinto, 2015).

(c) Use Time Ranges and Probability

Another Agile planning practice is to plan in ranges instead of specific dates. Traditionally, project managers look to predict the future based on deterministic estimations. When talking about a knowledge work environment, this approach is hugely flawed as things move quickly, and the work is rarely the same. Therefore, promising that a deliverable will be done by a specific date isn't realistic. To tackle this problem, Agile teams use time ranges and attach a specific probability based on their historical data. For example, it's much more realistic to say that a feature will be done between 5 and 8 days with an 85% probability than to make exact date promises (Noguera *et al.*, 2018).

To put this approach into practice, you can use a forecasting tool such as a Monte Carlo simulation. Based on a statistical analysis of your past performance, the tool can be very useful in determining the probability with which a project or a deliverable will be completed at a certain point in time. Here, the primary key to success when using Monte Carlo simulations is your workflow's stability.

(d) **Enable Transparency by Connecting Planning and Execution**

Last but not least, transparency should be introduced in the whole project management process by bringing together planning and execution. This will ensure visibility of the whole picture, the progress achieved is easily tracked, and reaction to changes happen on time. To do this in practice requires connection of multiple workflows. For example, there could be a workflow where plans of your projects are created and one where to visualize the individual tasks that every team member is responsible for (Nguyen, 2016).

Of course, the structure might be completely different based on the process and project scope. Nevertheless, what's important is the idea of full transparency in the process and measuring plans against their actual execution in real-time. As a result, there should be an open environment and improvements on project collaboration between team members and other relevant stakeholders.

Agile Best Practices for Work Execution

This section will discuss some of the best Agile practices related to executing projects in a knowledge work environment.

(a) **Visualize Workflow:** Even though workflow visualization isn't something hard to accomplish, it is one of the most effective Agile practices. Unlike manufacturing, knowledge work is practically invisible, so it's very hard to keep track and measure process efficiency. That's why simply visualizing your work stages and items will enable faster reactions to emerging issues and improve the collaboration between team members. To do this in practice, Kanban boards can be used. A good practice is to apply the lean management technique "value stream mapping" to see how the team delivers customer value. With this information in mind, optimization of workflow can be initiated by removing wasteful activities, re-structuring board based on customer needs, spotting and alleviating bottlenecks, etc.

Another great thing about visualization, and specifically Kanban boards, is that it can be applied across the entire company - from the management of multiple projects to multiple team's workflows. This will allow transparency at scale, which is one of the main themes when building an Agile organization. Moreover, by connecting all of these boards, it fosters an organizational structure where every layer regularly feeds with information the one above it or vice versa. As a result, whenever any emerging and unexpected changes occur, there will be a management system in place to take any necessary actions as soon as possible (Papadopoulos, 2015).

(b) **Limit WIP and Manage Queues:** WIP stands for work in progress and is one of the essential measures in the Agile world. It represents work that has been started but not finished yet, which means that it has not accumulated any outcomes. Therefore, having a lot of works in progress is a waste of time. That's why a useful Agile practice that comes from Kanban is to limit WIP to reduce waste and speed up the flow of the process. Applying this is extremely important in a knowledge work environment because it prevents context switching and enables team members to focus on finishing work instead of always starting a new one. Limiting WIP happens easily with the help of a Kanban board (Leite & Braz, 2016).

In addition to that, another practice that successful Agile teams use is to manage queues in their process, instead of tracking timelines. If you think about it, this makes perfect sense because by making sure that long queues don't form. Agile project managers can then reduce the cycle time of work items. Eventually, this automatically gives them control over timelines. An excellent tool for managing queues is the application of WIP limits. You can restrict how many items can enter a waiting stage to form a pull system in the process, where a task is started only when there is capacity available. As a result, this will lay the foundations of a stable and predictable value delivery process.

(c) **Reduce the Batch Size of Work Items**

Working on small batches is another critical Agile practice when executing projects. In fact, this is what enables the concept of frequent and continuous delivery in Agile project management. Reducing batch sizes also helps teams decrease the possibility of having many work items linger in the work process for too long. This facilitates flow and enables timely customer delivery of what has been promised. As a rule of thumb, a very small batch size that doesn't produce value is also not preferable. The idea here is to find the sweet spot and determine what the smallest possible work item that can be successfully released for customer examination is (Krehbiel *et al.*, 2017).

➤ **AGILE BEST PRACTICES FOR MONITORING**

The section consists of Agile practices for monitoring projects. This point is an integral piece of the puzzle where Agile teams synchronize progress and seek continuous improvement (Inayat & Salim, 2015). Some of the best Agile monitoring practices are summarized below.

(a) **Synchronize Progress Daily:** One of the most common Agile practices is to daily synchronize progress on projects with other team members. This usually happens due to the daily stand-up meeting where the entire team stands in front of a physical or digital Kanban board (or Kanban software) and engages in discussions about everything that has transpired since the last meeting. The main objective here is for everybody to give a quick status update, so there is a mutual awareness of who's working on what and

how the team is tracking against the plan. Furthermore, team members might need to escalate impediments to their flow in this meeting, so necessary actions could be undertaken as fast as possible (Paredes, 2024).

As a rule of thumb, the daily stand-up should be kept on point and below 15 minutes. There is no need for an extensive status report or engagement in lengthy discussions about emerging issues. The idea here is that progress is synchronized as quickly as possible and members quickly return to their work. In case some team members have issues, they should discuss it with a senior colleague privately outside the meeting. Agile ceremonies aim to create an information flow of internal and external feedback collected from all teams. Holding them regularly helps teams and organizations react fast to clients' and business changes.

(b) Track and Measure Flow Metrics: Flow is the pinnacle to a stable process and predictable project delivery, so measuring flow metrics in Agile is another best practice. Some of the most important flow metrics to track are WIP, cycle time, and throughput. Together, they will allow determination of what can be done in the process and how long it will take, so demand can better be matched with existing capabilities. For example, with the help of charts such as the cumulative flow diagram, cycle time scatterplot, heat maps, etc., the team will be equipped with the tools to measure the performance of flow and collect historical data. As a result, this will help to continuously track the demand and capacity levels in the system, so that predictable workflows created and realistic forecasts for project delivery can be given to customers (Rigby *et al.*, 2018).

(c) Engage in Frequent Reviews: Lastly, it's all about continuous improvement. That's why another helpful Agile practice is to implement regular feedback loops for reviewing work processes. In Kanban, for example, there is a regular service delivery review, which is usually held weekly or bi-weekly. There, teams reflect on all the deliverables that they have released to the customer and incorporate both internal and external feedback. This iterative process allows teams to continuously improve processes and quickly implement valuable lessons learned while the project is still active. As a result, they will be able to remain agile to any emerging changes that frequently meet customer expectations (Alsari *et al.*, 2020).

RECOMMENDATIONS:

- ✓ Adopt the Agile Methodology that Suits You
- ✓ Choose the Right Team Members
- ✓ Conduct Daily Meetings
- ✓ Use the Right Set of Professional Agile Tools
- ✓ Form Clear Goals and Responsibilities
- ✓ Promote Transparency among Team Members
- ✓ Foster a Culture for Learning and Improvement

Conclusively on this note, Agile teams must constantly seek ways to enhance their processes and increase efficiency in delivering value. Encouraging the team to refine and develop new skills is critical to this continuous improvement process. It is equally important for team members to remember the fundamental Agile principle of regular self-reflection. By regularly assessing their performance, the team can identify areas for self-improvement and adjust their approach accordingly (Weichbroth, 2022). This iterative approach enables the team to optimize their performance and deliver exceptional results continuously.

CONCLUSION

In the world of Agile project management, there are numerous practices that one can apply to improve the agility of a company's processes. However, a project manager should remember that Agile is a mindset and a unique approach to project management that doesn't prescribe a single best way to do it. That's why when talking about best Agile practices, they should only be classified as guidelines for transforming project management processes. The way they are applied depends on the specific situation. After all, adopting agility is a never-ending process, so it should be continuously experimented and the best solutions that work selected in the end.

REFERENCES

- Alsari, A., Qureshi, R., & Algarni, A. (2020, October). Agile framework to transform traditional team. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE.
- Anthony Jnr, B. (2023). Agile software development and software practitioners' productivity amidst the COVID-19 pandemic: a narrative review. *Journal of Science and Technology Policy Management*.
- Ersoy, I. B., & Mahdy, A. M. (2015). Agile knowledge sharing. *Int. J. Softw. Eng.(IJSE)*, 6(1), 1-15.
- Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2020). From agile to DevOps: Smart skills and collaborations. *Information Systems Frontiers*, 22(4), 927-945.
- Inayat, I., & Salim, S. S. (2015). A framework to study requirements-driven collaboration among agile teams: Findings from two case studies. *Computers in Human Behavior*, 51, 1367-1379.

- Junker, T. L., Bakker, A. B., Gorgievski, M. J., & Derks, D. (2022). Agile work practices and employee proactivity: A multilevel study. *Human Relations*, 75(12), 2189-2217.
- Khan, R. A., Abrar, M. F., Baseer, S., Majeed, M. F., Usman, M., Ur Rahman, S., & Cho, Y. Z. (2021). Practices of motivators in adopting agile software development at large scale development team from management perspective. *Electronics*, 10(19), 2341.
- Krehbiel, T. C., Salzarulo, P. A., Cosmah, M. L., Forren, J., Gannod, G., Havelka, D., ... & Merhout, J. (2017). Agile Manifesto for Teaching and Learning. *Journal of Effective Teaching*, 17(2), 90-111.
- Leite, M., & Braz, V. (2016). Agile manufacturing practices for new product development: industrial case studies. *Journal of Manufacturing Technology Management*, 27(4), 560-576.
- Malik, M., Sarwar, S., & Orr, S. (2021). Agile practices and performance: Examining the role of psychological empowerment. *International Journal of Project Management*, 39(1), 10-20.
- Masood, Z., Hoda, R., & Blincoe, K. (2018). Adapting agile practices in university contexts. *Journal of Systems and Software*, 144, 501-510.
- Mihalache, A. (2017). Project management tools for agile teams. *Informatica Economica*, 21(4), 85-93.
- Moloto, M., Harmse, A., & Zuva, T. (2020). Impact of agile methodology use on project success in organizations-a systematic literature review. *Proceedings of the Computational Methods in Systems and Software*, 267-280.
- Nguyen, A. (2024). Implementing Agile team structure for enhanced team performance. Retrieved from: www.timedoctor.com
- Nguyen, D. S. (2016). Success factors for building and managing high performance agile software development teams. *International Journal of Computer*, 20(1), 51-82.
- Noguera, I., Guerrero-Roldán, A. E., & Masó, R. (2018). Collaborative agile learning in online environments: Strategies for improving team regulation and project management. *Computers & Education*, 116, 110-129.
- Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia-Social and Behavioral Sciences*, 175, 455-463.
- Paredes, R. (2024). Getting started with Agile methodologies. Retrieved from: www.safetyculture.com
- Parker, D. W., Holesgrove, M., & Pathak, R. (2015). Improving productivity with self-organised teams and agile leadership. *International Journal of Productivity and Performance Management*, 64(1), 112-128.
- Rigby, D. K., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard business review*, 96(3), 88-96.
- Rigby, J.S. & Hirotaka, T. (2016). Embracing agile: How to master the process that is transforming management. *Harvard Business Review*, 40-48. Retrieved from: www.hbr.org
- Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International journal of project management*, 33(5), 1040-1051.
- Shirokova, S., Kislova, E., Rostova, O., Shmeleva, A., & Tolstrup, L. (2020, November). Company efficiency improvement using agile methodologies for managing IT projects. In *Proceedings of the International Scientific Conference-Digital Transformation on Manufacturing, Infrastructure and Service* (pp. 1-10).
- Waldron, K. (2017). Implementing a collaborative working environment using agile: the lexisnexis experience. *Legal Information Management*, 17(1), 16-19.
- Weichbroth, P. (2022). A case study on implementing agile techniques and practices: Rationale, benefits, barriers and business implications for hardware development. *Applied Sciences*, 12(17), 8457.
- Yu, X., & Petter, S. (2014). Understanding agile software development practices using shared mental models theory. *Information and software technology*, 56(8), 911-921.
- Žužek, T., Gosar, Ž., Kušar, J., & Berlec, T. (2020). Adopting agile project management practices in non-software SMEs: A case study of a Slovenian medium-sized manufacturing company. *Sustainability*, 12(21), 9245.