iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81975

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Factors Influencing Automated Agile Teams

Mohammed Mohiseen Ahamed[1], O Nagaraju[2]
[1]*Research scholar,* [2]*Research Supervisor, Computer science &Engg, Nagarjuna University, Guntur, India*

Abstract- As we discussed in the previous papers Automated Agile Virtual Teams practice and in order to create high customer satisfaction, increase in value product in business, agile method has made with many changes in management. It presents an automated agile virtual team which is automatic with using a concept virtual governor which is efficient, cheaper, fast, highly predictable, quality improvement and high customer satisfaction increases when compared to previous technologies. Now in this paper in order to support and enhance Automated Agile teams we are including the Factors influencing Automated Agile Teams that will improve performance and predictability which is possible with Automated agile method instead of traditional Agile method. In traditional method, low team autonomy and high individual. But Automated Agile method includes both individual and team autonomy. For faster, cheaper, continuous delivery, better software development, agile method is used.
Keywords: Agile, Virtual, Factors.

I. INTRODUCTION

Many software organizations are adopting/adopted Agile development methodologies rather than Traditional methodology due to increase popularity in agile method. In traditional software development teams are composed of individual with different roles like testers, developers, business analyst, designer etc. These roles are well defined, the team member function with their separate roles to be carried out within the boundaries. On other hand Agile Methodology in which individuals mange their own work, based on need they shift work among themselves and participate in team decision making. The work begins with set of requirement in documentation and followed by high level design development in traditional methodologies. On other hand, in Agile methodology it is easy. In agile methodology if any change occurs, team does not freeze the work where as in traditional methodology freeze requirement and disallow the change. In traditional methodology developer want a detailed specification because methodology assumes that customer do not know the requirement where as in Agile methodology both customer and developer learn together.

II. LITERATURE REVIEW

The Comparison between Traditional software development and Agile methodologies is given by (Mahdi Javanmard,Maryam Alian, 2015)[4]. The difference between Traditional software development and Agile methodologies is adaptability for project changes. The traditional methodology is static where as in agile methodology is adaptable for the changes in the project.
Comparison of different Agile methodologies like SCRUM, XP , feature driven development, dynamic system development method with general features is also given in it. Agile is people focused and adoptable instead of being predictive. It advocates a combined and small team that work closely together. But the team size is a factor is in turn controlled by people factors. When adopting Agile team to be effective, people factor is important(VikashLalsing , SomveerKishnah and Sameerchand Pudaruth, 2012)[1].Interaction among team members to distribute knowledge is well known in Agile software development . However, little assistance is provided to encourage interaction across agile teams. The paper examines influencing factors, such as organizational strategy, and communication flow and channels, regarding inter-team knowledge sharing (KS) effectiveness in agile environments(Viviane Santos, Alfredo Goldman,2014)[3]. A tool is proposed for foster reflection in Agile software development. A five factor model is contributed. To test this tool a empirical validation tool with scrum team is executed(Christoph J. Stettina and Werner Heijstek,2011)[2].

III. METHODOLOGY

The methodology describes the two critical environmental factors influencing automated Agile teams, that emerged from this research. These factors are: senior management support and level of customer involvement. First, senior management support is discussed in terms of (a) how senior management influences automated Agile teams and (b) how senior management support can be secured for the establishment, functioning, and propagation of these teams. Second, the level of customer involvement is discussed in terms of (a) how different levels of customer involvement influence automated Agile teams and (b) how satisfactory customer involvement can be secured for the smooth functioning of these teams.

©IJRASET: All Rights are Reserved

Finally, both the factors—senior management support and level of customer involvement—are discussed in light of existing literature.

A. *Influence of Senior Management Support*

Automated Agile teams are greatly influenced by the senior management at their own organizations The following sections describe the influence of senior management on automated Agile teams, followed by the strategies used by automated Agile teams to secure senior management support at their own organizations.

Senior Management Support Influence are Organizational Culture, Negotiating Contracts Financial Sponsorship, Resource Management. Securing Senior Management Support via Business Drivers Applicability to Project Context, Time to Market, Customer Demands ,Process Improvement.

Influence of Senior Management Support

- Organizational Culture
- Negotiating Contracts
- Financial Sponsorship
- Resource Management

Securing Senior Management Support via Business Driver

- Applicability to Project Context
- Time to Market
- Customer Demands
- Process Improvement

Senior management influences organizational culture, the types of contracts governing projects, financial sponsorship, and resource management. A senior management that does not support automated Agile teams might face several challenges for the team in each of these areas.

Organizational Culture. Organizational culture has been defined as “a standard set of basic suppositions invented, discovered or developed by the group when learning to face problems of external adaptation and internal integration”. Organizational culture has a strong influence on the ability of an agile team to be automated. Traditional software development teams typically adopt strictly hierarchical organization structures. Automated Agile teams on the other hand, require organization structures that are informal in practice, where the boundaries of hierarchy do not prohibit free flow of information and feedback. In an informal organizational structure, the senior management is directly accessible by all employees (maintaining an ‘open-doors’ policy), and accepts feedback—both positive and negative. Teams operated in Agile organizations using agile software development, are characterized by informal organizational structures. Informality in organizational structure promotes openness. Openness was one of the most common traits mentioned by participants, that made the organizational culture conducive for Agile teams. In such organizations, team members are free to voice opinions, raise concerns, seek management support in resolving their concerns, make collaborative decisions, and adapt to changes in their environment. This freedom provided by senior management is crucial for the team to achieve and sustain autonomy Starting with an informal structure has a cascading effect. Informality in the organizational structure leads to openness marked by free-flow of communication and feedback, which in turn leads to an organizational culture of trust. An organizational culture where teams trust their senior management to support them, and when senior management trusts the teams to perform and display responsibility, makes for fertile grounds for Automated Agile Teams to emerge.

In contrast, an organization with a strict hierarchical structure is not conducive for automated Agile teams. A common example is that of a government sector organization, with a strict hierarchical structure. The software development teams in such organizations form one of the lowest levels of hierarchy, topped by middle management, and then senior management. Such hierarchical structure is often coupled with heavy processes, such as heavy documentation, long change management processes, and long software delivery and deployment processes. Such a culture restricts both the team’s ability to practice light-weight Agile methods, and their ability to Automate. A strict hierarchical structure also has a cascading effect. The hierarchy in such an organization enforces a lack of openness marked by restricted and indirect lines of communication and feedback, which in turn leads to an environment of fear. Teams are afraid of voicing opinions, raising concerns, making collaborative decisions, and adapting to changes in their environment.

On the other hand, some government sector organizations find that their culture, while apparently different, can be receptive to changes brought on by Agile methods.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Senior management support, in terms of providing freedom and establishing an organizational culture of trust, is therefore extremely important for automated Agile to establish and flourish. A senior management that supports automated Agile teams will (a) maintain an informal structure, (b) provide freedom for teams to provide feedback, and (c) create an organizational culture of trust.

Negotiating Contracts. Automated Agile teams are influenced by the type of contracts that govern their projects. Senior management either directly in smaller organizations, or through their sales department in larger organizations—is responsible for negotiating contracts with customers. A customer can demand a fixed-bid contract where the cost, time, and scope of the project are fixed up-front. If senior management accepts the customer's demand for a fixed-bid contract, it has far-reaching consequences for the

Automated Agile team. Teams find that "fixed price doesn't work well with Agile" because "Agile talks about embracing change and can't do fixed price projects with changes coming in". The process of fixing the cost, time, and scope of the project in a fixedbid contract involves estimating the project. A senior management that does not support

Automated Agile teams, fixes the cost, time, and scope based estimates provided by managers, rather than the teams. As a result the team may be placed under pressure to deliver to often unrealistic estimates.

In contrast, a senior management that is aware of the negative consequences of fixed-bid contracts on the teams better supports Automated Agile teams. They provide customers with options. These options include offering an iteration on a trial basis, the flexibility to buy more iterations or terminate the contract with an iteration's notice, and swapping features.

By providing the customers with the option to quit the project in the worst case scenario, some of their financial risks are covered. So if the customers are unhappy with the results, they could always quit the project. If a customer is still insistent on a fixed-bid contract, the senior management can support a Automated Agile team by inviting the team to estimate their projects. Based on the rate of development per iteration the team velocity as a guideline, the team can estimate the time required for developing a particular set of requirements in a given domain. Then some amount of extra time could be added to the estimated time as a buffer. The contract is then drawn on this estimated time (including buffer) for a fixed price and scope.

A small amount of buffer time was important to allow the customer the possibility of introducing changes in requirements along the way while giving the development team time to respond to those changes. Buffering was a practical strategy of working with a fixed-bid contract while using Agile methods. Finally, senior management in Agile organizations are very careful about negotiating contracts that are "Agile-friendly". They frequently have a specialized sales team that understand Agile methods and the consequences of the contract on the Automated Agile teams.

A senior management that supports Automated Agile teams will (a) try to convince customers to try flexible contract options, (b) engage the team in providing estimates for the fixed-bid contract, along with adding a contingency buffer, or (c) negotiate "Agile friendly" contracts.

Financial Sponsorship. From the senior management, the Automated Agile teams need financial sponsorship in the form of Agile training and an infrastructure which is helpful to automated practices. The importance of a Mentor in the early stages of becoming a Automated Agile team is very important. In order to benefit in the form of an Agile Coach from the presence of a Mentor the team needs senior management support. The Agile Coach is often a contracting consultant, hired specifically to train a new team on Agile principles, values, and practices. In other cases, an existing project manager in the organization may take up the Mentor role. The senior management provides financial support by either hiring contracting Agile coaches or sponsoring these managers, and occasionally other team members, to receive Agile training (e.g. a Scrum Master Certification). Financial support is also required in the form of infrastructure support, such as setting up an open-plan workplace and tools for electronic communication and collaboration with distant customers. A supportive senior management champions the cause of Automated Agile teams and provides financial support for such an infrastructure.

A senior management that supports Automated Agile teams is willing to make such financial investments as (a) hiring a Mentor for new teams or providing existing Project Managers with Agile training and (b) providing the infrastructure necessary for effective functioning of the Automated Agile teams.

Resource Management. An important influence of senior management is the way they manage resources. For Automated Agile teams, dedicated resources are highly desired. When team members are allocated to multiple projects, it has a negative influence on the teams' ability to perform and automate. One of the main characteristics of Automated Agile teams is high levels of cohesion and collaboration within the team. The team's ability to Automate is dependent on understanding each others' strengths and weaknesses and forming a team culture of openness and respect. It takes time for a team to learn about each other and Automate based on the members' myriad abilities.

If the members are split across multiple projects, it affects their ability to perform group programming that enables Automated. A senior management that does not realize the implications of their resource management can have a negative influence on the team.

On the other hand, a supportive senior management values their teams and respects their human side as much, if not more, than their technical skills:

Resource management in terms of the hiring process and removal of individuals from teams is also influenced by senior management. In Agile organizations where senior management supports Automated Agile teams, their Human Resources department is set up specifically to hire people that are likely to "fit" into Agile teams. Sometimes, team members need to be removed from an Agile team because of their inability to fit into the culture. One of the team members typically takes on a Terminator role and seeks senior management support in removing such individuals. Senior management supports Automated Agile teams through managing resources by (a) providing dedicated resources to projects, (b) hiring individuals to fit into an Agile culture, and (c) removing individuals who threaten Automated teams with the help of a Terminator.

B. Securing Senior Management Support

While senior management support is extremely important for Automate Agile teams, it doesn't always come naturally. Supporting these teams involves the senior management changing their organizational culture, process of negotiating contracts, and resource management strategies. All senior management may not be ready to make such significant, organization-wide changes.

In non-Agile organizations, a pilot Agile team must secure senior management support in order to survive. One of the team members typically takes on the role of Champion to secure senior management support. It is important that the Champion understands their business drivers—the factors that motivate senior management's business decisions. Using these drivers, the Champion convinces senior management at their organization to support Automated Agile teams. Some of the business drivers or motivators include applicability of method to project context, time to market, cost-effectiveness, customer demands, and process improvement

Applicability to Project Context. From a senior management perspective, Agile methods are one of several methods from a tool-set that their teams can learn and use to better serve their customers. The applicability of a given method to a given project context is an important driver for senior management. Senior management typically remains open to various options that will bring good returns on investment:

A pragmatic Champion is aware that Agile methods may not be applicable to all types of project contexts. A Champion is cautious not to advocate Agile irrespective of project context an effort that can eventually backfire.

And so a Champion can advocate the use of Agile methods based on their applicability to the project context. They explain the advantages of Agile methods, given the organization's context. Senior management is much more likely to be convinced to invest in Automated Agile teams if they find that the practices fit the projects' contexts.

Time to Market. Time-to-market is another important driver for a senior management. In the present world of fast-paced development, cut-throat competition, changing customer requirements, and businesses thriving on innovation, the time it takes to develop and deploy a product to the market is an important driver for senior management. Faster time-to-market is one of the advantages of Agile methods showcased by a Champion in a bid to convince senior management. A Champion explains how Automated Agile teams are able to produce working software iteratively and incrementally such that changing customer requirements and latest business trends can be accommodated easily. They also highlight how Agile allows them to eliminate waste by focusing on customer priorities, which in turn leads to a shorter time to market.

Traditional hierarchical organizations often have heavy documentation processes. The time spent in lengthy up front documentation can be a huge waste in the face of changing requirements and can easily slow a product's time-to-market. When senior management realizes this problem, they are more willing to invest in Agile methods that offer just enough documentation and faster time-to-market. Fast time-to-market is one of the best cards a Champion can play, because faster delivery of working software is one of Agile's most commonly claimed advantages.

Customer Demands. While some customers are doubtful about Agile methods, other may specifically demand an Agile approach to developing their projects. With the growing popularity of Agile methods in software industries around the world, more customers are looking to engage in Agile projects.

Responding to their customers' demands is an important driver for senior management. As a result, they encourage their teams to learn Agile methods in response to customer demands. A related problem is that some senior management mandate an Agile approach in response to customer demands, but do not understand their own role in the process.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

A senior management that understands their own role in the process, not only mandates Agile projects in response to customer demands, but also changes their own practices to support Automated Agile teams.

Process Improvement. A choice of software development method can be driven by a need for process improvement. Senior management in organizations with no well-defined software development process are often easier to convince to try Agile. In contrast, senior management in organizations used to traditional software development methods, need to be shown a marked improvement brought on by the introduction of Agile methods. To convince senior management, the Champion collects and reports metrics to demonstrate process improvement. The metrics available in regular Agile projects are very different from those in non-Agile projects, so senior management can harbour misconceptions about the new metrics and struggle to understand them.

Senior management may relate formality with robustness, and so several Agile processes and artifacts that are paper-based and informal may appear less robust. An effective Champion understands the importance of translating Agile metrics into traditional metrics in the early stages of transitioning, so that their senior management can comprehend them and evaluate the performance of the pilot team.

Initial translation between Agile and traditional metrics allows senior management to ease into the process. While it takes time and effort to read and understand reports with two different metrics, it is a valuable long-term investment.

Senior Management is quick to spot processes that show marked improvement in team performance and electiveness. In the absence of any real drivers for change, senior management are not convinced about adopting Agile methods and making the organization-wide changes required to support Automated Agile teams.

C. Influence of Customer Involvement

Automated Agile teams are influenced by the level of customer involvement they receive on their projects. Inadequate customer involvement has negative consequences for the team, while adequate customer involvement has positive consequences for the team. This research found several influences of customer involvement on teams and multiple strategies for securing customer involvement. One way the customer's influence Automated Agile teams is through negotiating fixed-bid contracts. Customers demanding fixed-bid contracts place limitations on the team's ability to respond to changes. The main influence of negotiating contracts is a pressure to over commit. The influence of negotiating fixed-bid contracts and the strategies of providing flexible contract options and buffering have been discussed. The following sections describe the most critical influences of customer involvement on Automated Agile teams, followed by the some of the most popular and innovative strategies used by the teams to secure customer involvement. The emergence of level of customer involvement from the underlying concepts. The below table presents an overview of the influence of customer involvement and the various strategies used by Automated Agile teams to secure adequate involvement discussed in detail.

Influence of Customer Involvement
Gathering and Clarifying Requirements
Prioritizing Requirements
Securing Feedback

Securing Customer Involvement
Changing Mind-set
Changing Priority
Story Owners
Just Demos
e-Collaboration

Gathering and Clarifying Requirements. Customer representatives are meant to provide requirements in the form of user stories every iteration. They are also responsible for clarifying these stories for the development team as needed. In real-life agile projects, however, development teams faced challenges in retrieving requirements from customers.

Similarly, some teams have issues trying to get customer representatives to clarify requirements. Without clear requirements, teams are forced to make assumptions about the customer's needs and priorities. These inaccurate assumptions lead to the team building features that are not as per the customer's intended requirements. The teams would then have to perform rework which incurs additional costs for the customers.

Rework is both costly to customers and taxing for developers if it has to be done at a later time. Due to delays in customer feedback, the need for rework typically does not surface until much later, by which time it is difficult for the developers to return to a particular story and rework it. As a result of insufficient and ineffective customer involvement, the development teams were unable to get customer representatives to provide and clarify requirements.

Prioritizing Requirements. Agile methods require customer representatives to prioritize the order in which the team should work on the user stories, driven by business value. Understanding and using the concept of prioritization doesn't always come naturally to customers new to Agile projects.

Some teams face difficulties in getting customer representatives to prioritize the requirements and as such the teams are confused about what features to develop and deliver first. In contrast, teams that receive adequate levels of customer involvement are able to gather and clarify requirements from their customers more easily and effectively.

Securing Feedback. Customer feedback is of vital importance in ensuring the desired product is being developed and delivered incrementally. As a senior developer pointed out "the whole point of the two week iterations was so that the end users could know if we were on the right track" and requires the customer representatives to provide feedback on developed features. In absence of customer feedback, some teams are unable to assess how well the features meet the requirements. In contrast, teams that receive adequate levels of customer involvement have better, more direct, and more frequent communication with their customers.

D. Securing Customer Involvement

Several interesting strategies are used by the teams to secure customer involvement. These strategies are collectively named Agile Undercover, a category that emerged from the data analysis . These strategies include: Changing Customers' Mind-sets, Providing Options, Buffering, and Changing Priority, Risk Assessment Up Front, Story Owners, Using a Co-ordinator, Using a Translator, Just Demos, E-collaboration, and Extreme Undercover. Agile Undercover strategies allow teams to successfully secure customer involvement in some cases and continue to practice Agile in the face of inadequate customer involvement in others. Risk Assessment Up Front is a general strategy for assessing the risks involved in an Agile project up front. The level of customer involvement was one of the risk items assessed using a risk assessment questionnaire.

Extreme Undercover was a strategy used by some teams to practice Agile internally while appearing to be a traditional software development team to the customers. This strategy was found to be used by some Indian teams in the initial stages of adoption, where they faced extremely sceptical customers. Over the course of the research, as the popularity of Agile methods increased, this strategy was rarely observed. Providing options and buffering have already been described and also presented. A Translator was used to overcome the language barrier between teams and their customers. A Co-ordinator was used to help co-ordinate customer collaboration and change requests across distances.

Changing Mindset. A Promoter's role in convincing customers to try Agile methods and collaborate with teams is extremely important. Some customers harbour scepticism about Agile methods and are unwilling to extend collaboration. A Promoter tries to change the mind-set of such customers by explaining the principles and values of Agile methods. One of the participants, who played a Promoter role, highlighted the advantages of Agile methods to their customers in a bid to secure their involvement.

A Promoter asserts that frequent customer involvement allows customers better control of the product. A constant focus on customer priorities was seen as an advantage by customers, many of whom became willing to get involved in the process. Finally, in a bid to change customers' mind-sets, some Agile organizations offer Product Owner training to their customers in order to familiarize them with their responsibilities as an Agile customer.

Changing Priority. In an effort to maintain the iterative and incremental nature of their Agile projects, teams are forced to lower the priority of user stories that are awaiting customer requirements, clarification, or prioritization. Such stories are usually demoted in priority and pushed further down into the product backlog until the required customer response is secured and development on those stories can re-commence. Agile teams confess that they change the priority of the story in absence of enough, clear, and prompt requirements.

A user story was considered ready when the customers had provided the business goals and expected outcome associated with the story and implementation details necessary to estimate the story had been discussed. A story that was not ready was not able to achieve priority in the product backlog.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Story Owners. In absence of the on-site customer, agile teams use Story Owners where members of the customer organization share the responsibility of the customer role and are available as and when required. The practice of assigning Story Owners was an adaptation to the Scrum practice of allocating a product owner. Story owners are responsible for particular stories (less than a week long), instead of all the stories in the product backlog. Assigning story owners serves a three-fold purpose. Firstly, having multiple story owners instead of a single customer representative for entire project means no one person from the customer's organization is expected to be continuously available. This lessens the burden of the customer representatives, who have their own operational jobs to tend to alongside playing an Agile customer.

Secondly, it allows the team to plan out stories for development in synchronization with the corresponding story-owner's availability. Thirdly, it encourages a sense of ownership among customer representatives as they are encouraged to present their own stories to peers at end of iteration reviews.

Just Demos. Demonstrations are used by agile teams as a powerful mechanism to secure the much needed and elusive customer feedback. The team presents working software to the customer representatives at these regular demonstration meetings and receive feedback from them regarding the features delivered in that iteration. This feedback is then incorporated into the development cycles.

Using demos provides the opportunity to clear any assumptions made by development teams as a consequence of the customer representative not providing enough or clear requirements

Demonstrations were often the only regular involvement that some Agile teams receive from their customer representatives. The teams use this opportunity to receive feedback and clarifications. The customers appreciate the demos despite their potential reservations about Agile in general because it provides them with increments of working software. Teams utilize demos to discuss requirements and get clarifications in addition to receiving feedback on demonstrated features.

E-collaboration. Electronic collaboration (e-collaboration) is a popular means of communicating with customers using phone, email, chat, and voice/video conferencing. With increasing number of software projects being off-shored globally or spread across multiple sites, face-to-face collaboration has become a practical challenge. E-collaboration is a popular alternative used by software teams to overcome the issue because (a) Agile requires regular customer involvement (b) several teams have physically distant customers making face-to-face collaboration difficult and (c) e-collaboration provides a cheaper alternative. In summary, adequate levels of customer involvement is extremely important for Automated Agile teams. Inadequate customer involvement leads automated Agile teams to adopt coping strategies, many of which are not ideal. Teams that receive adequate customer involvement, on the other hand, are able to concentrate on delivering quality products to meet their customers' demands.

IV. CONCLUSION

Teams depend on environmental factors such as the support of senior management at their own organization and the level of customer involvement on their projects. The Champion and Promoter roles, mostly played by Agile coaches, handled these relationships. Senior management influences the organizational structure and culture in an organization. The importance of senior management support in the form of a conducive organizational culture has been widely acknowledged. Agile methods challenge conventional management ideas, and require changes in organization structure, culture, and management practices in traditional software development organizations. Customer collaboration in traditional software development projects is typically limited to providing the requirements in the beginning and feedback towards the end, with limited regular interactions between the customer and the development team. In contrast, customer collaboration is a vital feature and an important success factor in Agile software development. Agile methods expand the customer role within the entire development process by involving them in writing user stories, discussing product features, prioritizing the feature lists, and providing rapid feedback to the development team on a regular basis.

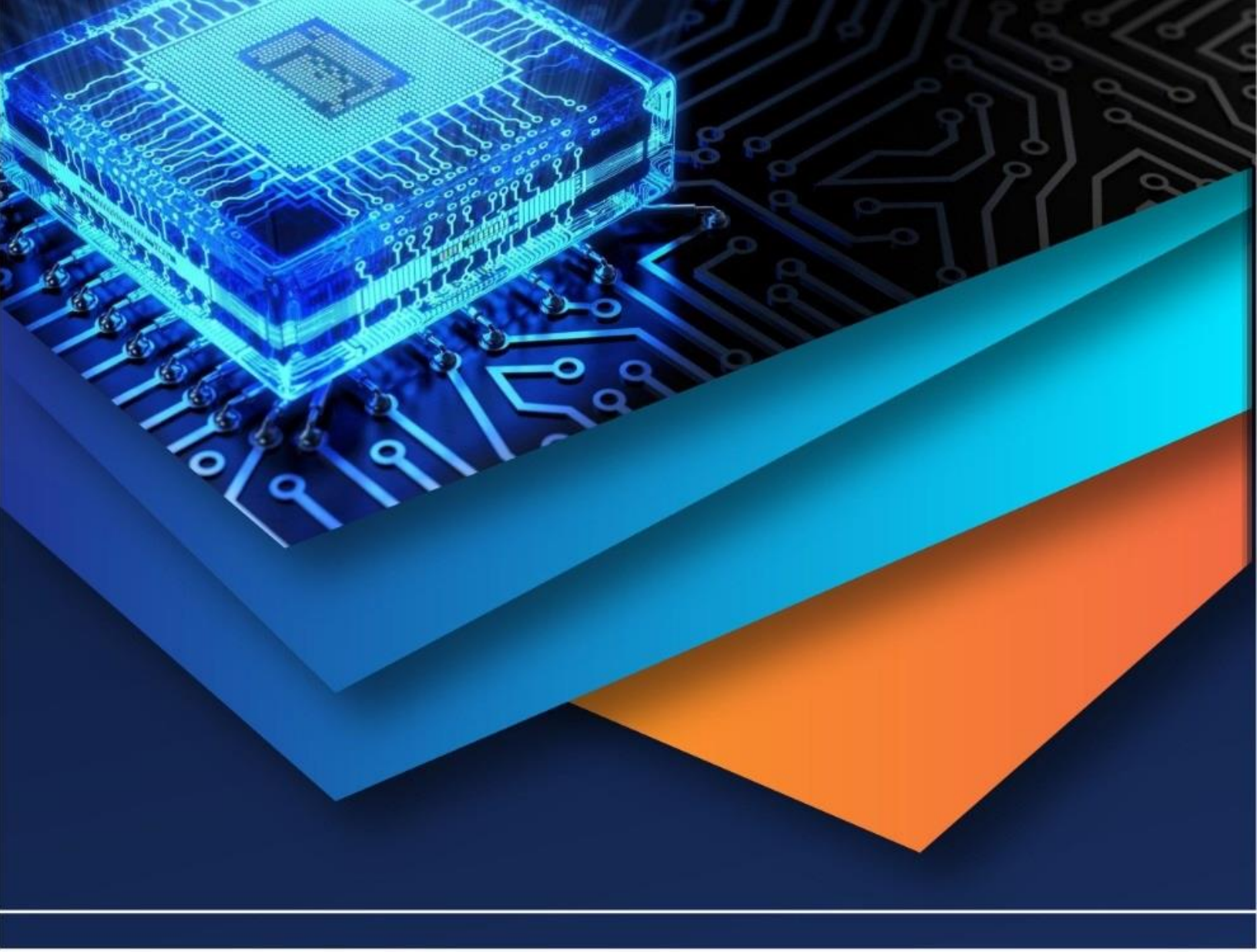
REFERENCES

[1] Vikash Lalsing, Somveer Kishnah and Sameerchand Pudaruth, People Factors In Agile SoftwareDevelopment and Project Management International Journal Of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012

[2] Christoph J. Stettina,Werner Heijstek, "Five Agile Factors: Helping Self-management to Self-reflect" EuroSPI 2011: Systems, Software and Service Process Improvement pp 84-96

[3] Viviane Santos, Alfredo Goldman,2014 " The Influence of Organizational Factors on Inter-team Knowledge Sharing Effectiveness in Agile Environments"47th Hawaii International Conference on System Sciences.

[4] Mahdi Javanmard, Maryam Alian, "Comparison between Agile and Traditional software development methodologies" Cumhuriyet University Faculty of Science Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015) ISSN: 1300-1949



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)