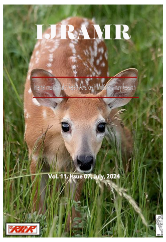
ISSN : 2350-0743

International Journal of Recent Advances in Multidisciplinary Research

Vol. 11, Issue 07, pp. 10124-10128, July, 2024



RESEARCH ARTICLE

AGILE LEADERSHIP AND THE PSYCHOLOGY OF FEEDBACK: CULTIVATING A CULTURE OF CONTINUOUS IMPROVEMENT AND QUANTITATIVE ANALYSIS BETWEEN AGILE AND TRADITIONAL LEADERSHIP

***Hilal Durak, Harun Gultekin, Luke Roumy, Frank Orlando and Chang-Woo Ham**

Software Engineering Program, School of Systems and Enterprises, Stevens Institute of Technology

ARTICLE INFO

Article History

Received 10th April, 2024
Received in revised form
16th May, 2024
Accepted 17th June, 2024
Published online 30th July, 2024

Keywords:

Recast, Error Correction, Indirect Feedback.

****Corresponding author:***
Hilal Durak

ABSTRACT

In today's dynamic and rapidly changing working environments, companies and organizations face the challenge of continuous adaptation to change in the industry. Agile methods have been growing rapidly among software development over recent years, and this article explores the relation between agile leadership in general and the psychology of feedback to foster a culture of continuous improvements. Through an examination of theoretical frameworks and empirical findings, this article underscores the importance of leadership behaviors that promote psychological safety, transparency, and empowerment in soliciting and utilizing feedback effectively. Ultimately, this article seeks to provide actionable insights for leaders and organizations seeking to navigate the complexities of change and drive sustained performance improvement in today's dynamic business landscape.

Citation: Hilal Durak, Harun Gultekin, Luke Roumy, Frank Orlando and Chang-Woo Ham. 2024. *"Agile Leadership and the Psychology of Feedback: Cultivating a Culture of Continuous Improvement and Quantitative Analysis between Agile and Traditional Leadership.", International Journal of Recent Advances in Multidisciplinary Research,* 11, (07), 10124-10128.

INTRODUCTION

The Rise of Agile Methodologies: In an era characterized by rapid technological advancements, global competition, and shifting consumer preferences, companies and organizations must continuously evolve and innovate to thrive. Traditional leadership models like the waterfall model, which requires careful planning and no room for failure, are often ill-equipped to navigate the complexities of today's business landscape with one journal stating "...the process was too rigid and that each step was siloed from the rest, creating inefficiencies and lengthening the development cycle" (McQuade, Hunter & Moore, 2019). To address the flaws of waterfall methodology, the Agile Manifesto was created to emphasize individuals and interactions, working software, customer collaboration, and response to change. Developed originally for software development, it is still predominantly used within the IT industry, but due to its success, it has now started to spread to non-IT projects.

Agile Frameworks and Roles: Agile is an overarching term for the principles and methodologies defined by the Agile Manifesto and its core principles. When looking into the actual practice of Agile software development, numerous frameworks have been developed to employ Agile methodologies into real world applications. One popular framework, Kanban, prioritizes the visual representation of workflow among teams. The kanban board provides a digital tool to trace workflow, enhance collaboration, and broaden accessibility to all team members. This framework focuses on continuous improvement by creating feedback loops where changes are introduced, making changes collaboratively and involving all stakeholders as needed. By following this model, boundaries are created "regarding the amount of work that is to be done or completed, so that the team doesn't over-commit additional work" and the workflow reduces anything considered "wasteful" (PremierAgile, par 6). In Kanban, there are no set roles that team members play so collaboration is a key characteristic that must be shared among all members within the team.

Like Kanban, Scrum is another framework that is widely used amongst teams in many different business pursuits. In this framework, different members of the team have specified roles. These roles include: Product Owner, Scrum Master, and the Development Team. The Product Owner is the bridge between the business and the stakeholders, meaning that they have the most amount of communication with who the product is for. They help manage the project's requirements using the product backlog and release updates in sprint reviews for "real customer usage and feedback" (West, par 20). The scrum master is "responsible for gluing everything together and ensuring that scrum is being done well" through communications with the product owner and the development team (West, par 22). With the product owner, the scrum master manages the product backlog and helps the owner plan out the project to deliver the most effective product. They also help the product owner plan for sprints and sprint reviews to keep the team focused on the end goal. For the development team, the scrum master keeps them organized by managing blockers in order to effectively work through each sprint. The development team consists of a group of people who have separate roles. Some team members could be designers, writers, or the developers themselves. The development team is responsible for delivering the planned work at the end of each sprint, and meet during the daily scrum meeting (i.e. standup) to "provide transparency to the work", especially to "seek help, talk about success and highlight issues and blockers" (West, par 13). Scrum is a framework that is focused on learning through experiences and making changes along the way. The cadence of fixed-length sprints allows anyone following this framework to return to and reflect on the work that had been completed during the sprint reviews, daily scrum and the sprint retrospective. If scrum is the chosen framework, trust is the most important characteristic that each member of the team needs to have, as each person has a separate responsibility and any missing part puts the whole project behind (scrum.org, par 5). Besides, agile frameworks, with a focus on understanding their impact on user engagement and satisfaction of team members (Durak& Gultekin, 2024).

One other agile framework that works well with smaller and fast moving/changing projects is Extreme Programming (XP). XP "is used to improve software quality and responsiveness to customer requirements" because of the many practices that it follows (GeeksForGeeks, par. 1). Some of the practices that are followed in this framework are testing and integration testing to remove errors and find bugs during the development stage. Paired programming is a method that can be followed in order to limit the number of bugs and keep pushing on progress. Incremental development and small releases are helpful practices to gather insight from customers about the project and its progress. One practice that improves the readability and design of the code is refactoring; it enables the code to be easily maintained without changing the functionality of it. That means that within this framework, there is a collective ownership of the code and "all team members are responsible for the code, and anyone can make changes to any part of the codebase" (GeeksForGeeks, par 47). The development process that XP follows starts XP Planning, then XP Design, XP Coding and finally, XP Testing. In planning, user stories are created and an estimate on cost and the project's velocity are made. In design, prototypes are designed and class responsibility collaboration is encouraged. For coding, unit tests are developed before any code is actually written and paired-programming is a practice that is widely used. Finally, in testing, unit tests are executed daily to ensure the team remained on track with the requirements stated by the customers. Like scrum, XP has roles designated to members of the team. There is the developer, the customers, the manager, and the coach. As the developer, responsibilities include estimating the time and cost of the development process, write/perform unit tests and code parts of the project and refactor code. As a customer, it "is required to be in constant communication with the team and speak as a single voice to the team" in order to set priorities and explain expectations for the user stories (TutorialsPoint, par 7). The manager plans and monitors the planning game and keeps track of the progress being made through functional testing and metrics gathering. The coach plays a major role to teams who are new to extreme programming. As a coach, responsibilities include being able to apply XP practices to the project, and being readily available to help the team if any problems arise. While these are all popular methodologies that are widely used, there are many differences between them that are important to note in order to determine which one is best for a project. Kanban is a good choice when it is easier to visually see the progress of a project. Projects that might use Kanban are ones that have "widely-varying priorities" because they allow for changes to be made to a project "mid-stream" (planview.com), similar to extreme programming. Unlike Kanban, scrum is best used for teams that have "stable priorities". Because everything is planned out into sprints prior to the start of development, "changes during the sprint are strongly discouraged" and are meant to be reviewed at the end of a sprint (planview.com). Not only that, but scrum, as well as XP, have predefined roles for members on the team, while Kanban does not have designated roles. If teams are working on smaller projects or more fast-paced projects, extreme programming might be a better option because the team develops for about 1-2 weeks, whereas, in scrum, sprints can last for 2 weeks to 1 month. Not only that, but in XP the customer has the most say on the priorities of the project, but in scrum the product owner does, making customer involvement more important in XP (GeeksForGeeks). Regardless of the project, it is crucial to determine which framework to use in order to properly execute the project based on the needs of the team, the business and the customer.

Agile Leadership: Agile leadership is a departure from conventional hierarchical structures, with a shift towards traits like adaptability, collaboration, customer value and empowerment. Leaders in Agile environments are servant leaders and guide their team with a focus on flexibility, iterative progress, and continuous improvement through feedback. But to be an effective leader who embraces these focuses, the agile leader should be:

- An effective Communicator who fosters and increases trust, is transparent, and considers differences. Knowing that according to the human psychology, human beings create their own world and have different worlds from others (Cömertpay&Durak, 2023).
- A diplomat who considers the impacts on all stakeholders and how to follow up with all those affected.
- An effective listener who checks that they understand the meaning being portrayed.

- An Analytical thinker who must be able to anticipate outcomes, problems and solutions. (Crowder, Freiss 2015)
- In order for Agile to thrive the leader must help establish a foundation of trust. Feedback without established trust renders the feedback as meaningless as it "...fosters resentment and breeds an attitude of lack of commitment to the program/project". If you were to receive feedback from someone you do not trust or respect, then this feedback will lose all meaning.

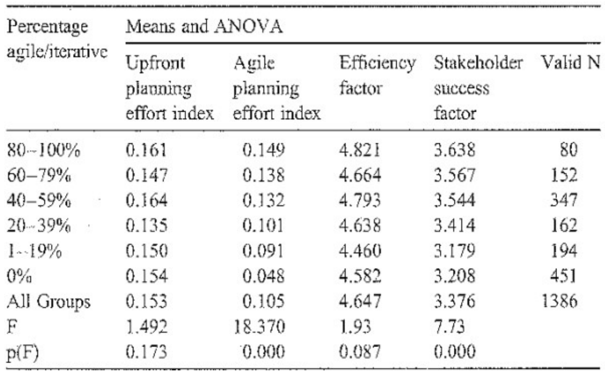
Trust and respect form the foundation, only then meaningful feedback can be built. It's easier to foster a sense of shared responsibility among team members when all of these things are in order.

Psychology of Feedback: Feedback is "...conceptualized as information provided by an agent (e.g., teacher, peer, book, parent, self, experience) regarding aspects of one's performance or understanding" and "is a 'consequence of performance'" (Hattie, Timperley. 2007). At the heart of it all, agile leadership is dependent on feedback to encourage and nurture an environment of continuous improvement. As emphasized by Kortum "it is not only essential to identify weak performances, but also to characterize the primary cause to solve the underlying problems" (Kortum, Klunder & Schneider, 2019). It is not enough to simply pinpoint the shortcomings, but to identify the root causes. Once you have identified the root causes you will be able to grow and learn. A key component of agile is continuous improvement and this can only be achieved through learning from one's mistakes. Habiba Saedon conducted a study to analyze the relationship between feedback and intrinsic motivation. It was found that "good feedback can lead to increased motivation and confidence in trainees. On the other hand, negative feedback is not aimed to demotivate or demoralize a trainee, but should be taken as constructive criticism for trainees to improve" (Habiba Saedon, 2012). Whether it is positive or negative feedback it is essential to be given, because those that receive positive feedback are more intrinsically motivated to continue improving. Those that receive negative feedback will know exactly what they have to do to improve, so it is important that people aren't afraid to give honest feedback.

To summarize, feedback allows for individuals to grow as it places the emphasis on the individuals in agile development rather than the process and tools used (Crowder, Freiss, 2015). Feedback not only helps an individual understand what they need to do to improve upon, but also their strengths and weaknesses. When you understand your strengths and weaknesses you are able to play into your strengths to benefit the entire team.

The Benefits of Feedback Oriented Leadership: Although Agile methods are continuing to gain popularity and are spreading beyond software development projects, little research has been done as to whether Agile projects truly are more successful, with the majority of existing research's methodology being anecdotal and not quantitative. Luckily, a study published in the International Journal of Project Management called *Does Agile work? - A quantitative analysis of agile project success* (Serrador, Pinto 2015) shows the impact of Agile methodologies on project success. This study, through rigorous quantitative analysis: surveyed Project management members from PMI and LinkedIn, gathered information on the projects they did and how successful they thought the project was, and whether these projects used agile or not. Although most of this data was self-reporting (subjective), the research found that there was a direct relation to agile approach and reported project success.

Table 01. Methodology type means with ANOVA analysis: how much of the project was done using agile or iterative techniques



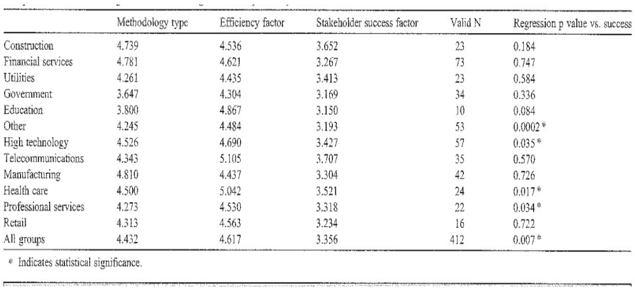
Percentage agile/iterative	Means and ANOVA				
	Upfront planning effort index	Agile planning effort index	Efficiency factor	Stakeholder success factor	Valid N
80–100%	0.161	0.149	4.821	3.638	80
60–79%	0.147	0.138	4.664	3.567	152
40–59%	0.164	0.132	4.793	3.544	347
20–39%	0.135	0.101	4.638	3.414	162
1–19%	0.150	0.091	4.460	3.179	194
0%	0.154	0.048	4.582	3.208	451
All Groups	0.153	0.105	4.647	3.376	1386
F	1.492	18.370	1.93	7.73	
p(F)	0.173	0.000	0.087	0.000	

As seen in this table, the researchers have found that the higher the percentage of the project was done with agile projects, the higher the Efficiency factor and Stakeholder success factor was.

Although the distribution of the project sample data was not evenly distributed, we see that compared to the projects that used 0% agile, the projects that used 20%+ agile methods had higher efficiency and stakeholder success numbers, showing that projects that use agile for most of their projects showed better results than the ones that didn't.

The Negatives?: In the same research above (Serrador, Pinto 2015), it was found that some industries performed better with agile while other industries performed more or less the same without agile.

Table 02. Comparison of means and regression results for agile success by industry



	Methodology type	Efficiency factor	Stakeholder success factor	Valid N	Regression p value vs. success
Construction	4.739	4.536	3.652	23	0.184
Financial services	4.781	4.621	3.267	73	0.747
Utilities	4.261	4.435	3.413	23	0.584
Government	3.647	4.304	3.169	34	0.336
Education	3.800	4.867	3.150	10	0.084
Other	4.245	4.484	3.193	53	0.0002*
High technology	4.526	4.690	3.427	57	0.035*
Telecommunications	4.343	5.105	3.707	35	0.570
Manufacturing	4.810	4.437	3.304	42	0.726
Health care	4.500	5.042	3.521	24	0.017*
Professional services	4.273	4.530	3.318	22	0.034*
Retail	4.313	4.563	3.234	16	0.722
All groups	4.432	4.617	3.356	412	0.007*

* Indicates statistical significance.

As seen in this table, the industries that are starred shows a significant when regressing methodology type vs over success measure. This highlights the fact that agile methodology was originally designed for high tech and IT fields like software and Health Care and these industries find most success using agile methods. On the other hand, industries like construction have higher Regression p value vs Success values, which shows that industries like Construction, Manufacturing and retail don't have a significant relationship between agile methods and success measure.

DISCUSSION

While the quantitative analysis conducted by Serrador and Pinto (2015) provides valuable insights on the importance of agile for a project's success, it is essential to recognize that success in agile goes beyond these numerical metrics such as efficiency and shareholder satisfaction. There are other qualitative factors that are just as important, like innovation, team satisfaction, and adaptability, which can also contribute to a project's success. The study highlights that the success of agile methodologies extends into areas often overlooked by traditional project management approaches. Agile frameworks, such as Kanban, Scrum, and Extreme Programming (XP), emphasize continuous improvement and learning through iterative processes. This iterative nature allows teams to adapt and evolve, fostering a culture of innovation and responsiveness that is crucial in today's fast-paced business environment. Moreover, the success of agile projects is not solely dependent on the framework employed but also on the quality of leadership within the organization. Agile leadership, characterized by adaptability, collaboration, and empowerment, plays a critical role in cultivating an environment where teams can thrive. Leaders must establish psychological safety, where team members feel safe to share ideas, take risks, and provide honest feedback. As noted in the attached article, feedback is vital for continuous improvement and intrinsic motivation. Effective agile leaders facilitate an open feedback culture, ensuring that feedback is constructive and aimed at both recognizing strengths and addressing areas for improvement. However, the transition to agile methodologies can present significant challenges, particularly for organizations entrenched in traditional hierarchical structures.

Aligning with agile's core principles, such as collaboration and adaptability, requires a fundamental shift in mindset and organizational culture. Strong leadership is essential in navigating this transition, as leaders must provide clear guidance, support, and continuous feedback to help employees align with agile principles. Furthermore, the study by Serrador and Pinto also points out that the success of agile methodologies varies across different industries. While agile has shown substantial benefits in sectors like IT and healthcare, other industries such as construction and manufacturing have not seen the same level of success. This discrepancy underscores the importance of context when implementing agile methodologies. Organizations must consider their unique industry characteristics, project requirements, and team dynamics when deciding whether and how to adopt agile practices. In conclusion, while the quantitative analysis underscores the positive impact of agile methodologies on project success, it is crucial to acknowledge the broader qualitative factors and the pivotal role of leadership in fostering a successful agile environment. The challenges associated with transitioning to agile highlight the need for strong, adaptable leadership capable of guiding teams through change and fostering a culture of continuous improvement and feedback. As organizations continue to navigate the complexities of the modern business landscape, the integration of agile principles, supported by effective leadership, will be key to sustained success and innovation.

REFERENCES

Clark, T. (2022, February 23). Agility from first principles: Agile doesn’t work without psychological safety. Harvard Business Review**.** https://hbr.org/2022/02/agile-doesnt-work-without-psychological-safety

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. Information Systems Research, 20(3), 329–354. http://www.jstor.org/stable/23015469

Cömertpay, E., &Durak, H. (2024). Examining the regulatory role of age in the relationship between existential anxiety and secondary traumatic stress in teachers. *International Journal of Education Technology and Scientific Researches***, 9**(26), 53-83. https://dx.doi.org/ 10. 35826/ijetsar.723

Crowder, J. A., &Friess, S. (2015). The psychology of agile team leadership. In Agile Project Management: Managing for Success (pp. 9-25).

Durak, H., Gultekin, H., Haralampopoulos, D., Skoulidas, L., Ciraulo, N., Vipparla, P. K., &Korczukowski (2024). User-centered design in agile: Integrating psychological principles for enhanced user experience. *International Journal of Recent Advances in Multidisciplinary Research,* **11**(05), 9888-9892.

Extreme programming - roles. (n.d.). https://www tutorialspoint.com/ extreme_ programming/ extreme_ programming_roles.htm

GeeksforGeeks. (2023a, February 22). Difference between Scrum and XP. https://www.geeksforgeeks.org/difference-between-scrum-and-xp/

GeeksforGeeks. (2023b, November 28). What is Extreme Programming (XP). https://www.geeksforgeeks.org/software-engineering-extreme-programming-xp/

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research, 77*(1), 81-112. https://doi.org/10.3102/003465430298487

Kanban vs. Scrum: What are the differences? (2021, October 13). Planview. https://www.planview.com/ resources/ guide/ introduction-to-kanban/kanban-vs-scrum/

Kortum, F., Klunder, J., & Schneider, K. (2019). Behavior-driven dynamics in agile development: The effect of fast feedback on teams. In 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP).https://doi.org/10.1109/icssp.2019.00015

Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. MIS Quarterly, 34**(**1), 87–114. https://doi.org/10.2307/20721416

McQuade, M. R., Hunter, A., & Moore, S. (2019). Appendix A: How agile software development came to enable adaptable systems**.** *In Acquisition of Software–Defined Hardware–Based Adaptable Systems* (pp. 31–36). Center for Strategic and International Studies (CSIS). http://www.jstor.org/stable/resrep22602.11

Rad, D., & Rad, G. (2021). Theories of change in agile psychology. *Technium Social Sciences Journal***, 21,** 570.

Rehkopf, B. M. (n.d.). Kanban vs Scrum. **Atlassian.**https://www.atlassian.com/agile/kanban/kanban-vs-scrum

Saedon, H., Salleh, S., Balakrishnan, A., Imray, C. H., &Saedon, M. (2012). The role of feedback in improving the effectiveness of workplace-based assessments: A systematic review. *BMC Medical Education*, **12**(1). https://doi.org/10.1186/1472-6920-12-25

Serrador, P., & Pinto, J. (2015). Does agile work? - A quantitative analysis of agile project success. *International Journal of Project Management***.** https://www. sciencedirect. com/science/ article/abs/pii/ S02637 8631 5000071

Top 7 agile frameworks. (n.d.). https://premieragile.com/types-of-agile-frameworks/

West, B. D. (n.d.). A deep dive into Scrum team roles. Atlassian. https://www.atlassian.com/agile/scrum/roles

What is Scrum? (n.d.). Scrum.org. https://www.scrum.org/ resources/what-scrum-module

****