

4-1-2022

Have the Agile Values endured? An empirical investigation on the 20th anniversary of the Agile Manifesto (2001)

Ashish Kakar
Texas Tech University, ashish.kakar@ttu.edu

Bismita Choudhury
Assam Downtown University, Bismitachoudhury@gmail.com

Akshay Kakar
West Virginia University, akshay.uh@gmail.com

Adarsh Kumar Kakar
Alabama State University, akakar@alasu.edu

Follow this and additional works at: <https://aisel.aisnet.org/sais2022>

Recommended Citation

Kakar, Ashish; Choudhury, Bismita; Kakar, Akshay; and Kakar, Adarsh Kumar, "Have the Agile Values endured? An empirical investigation on the 20th anniversary of the Agile Manifesto (2001)" (2022). *SAIS 2022 Proceedings*. 8.

<https://aisel.aisnet.org/sais2022/8>

This material is brought to you by the Southern (SAIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in SAIS 2022 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Have the Agile Values endured? An empirical investigation on the 20th anniversary of the Agile Manifesto (2001)

Ashish Kakar

Texas Tech University
ashish.kakar@ttu.edu

Akshay Kakar

West Virginia University
akshay.uh@gmail.com

Bismita Chaudhary

Assam Downtown University
bismitachoudhury@gmail.com

Adarsh Kumar Kakar

Alabama State University
akakar@alasu.edu

ABSTRACT

This study investigates whether the Agile Values introduced in the Agile Manifesto (2001) have endured today two decades later and whether they are still relevant to software developers. Further, are they positively correlated with work and affective outcomes of software development projects? We find out by conducting a survey with team members of 58 software development project in one of the largest global IT firms. To our surprise we find all the four Agile values have endured. The agile values still resonated with software developers. Additionally, overall, the values were positively correlated with team motivation, project effectiveness and project innovation. However, they were negative correlated with project efficiency and had no correlation with work exhaustion of team members. As expected, projects using Agile and plan-driven methodologies showed differential findings.

Keywords

Agile Manifesto, Core Values, Work Outcomes, Affective Outcomes

INTRODUCTION

For many decades, software engineering was focused on heavy-weight approaches aimed at success in developing increasingly complex business applications speedily, at lesser costs and of higher quality. Formal methods based on scientific management principles using a variety of tools and techniques for measurement and standardization of the software process were adopted in the belief that it would result in success in software development activities (Kakar, 2020). However, in the late 1990s, as disenchantment with the heavy-weight engineering methods grew, suggestions for improvement came from practitioners culminating in the Agile manifesto (Fowler and Highsmith; 2001). The Agile Manifesto caught on quickly with the software development community. By 2007 84% of the respondent organizations were using agile methods within their organizations which rose to an impressive 97% by 2018 (Hoda, Salleh and Grundy, 2018).

The Agile manifesto consists of the Agile Values and the Agile principles. The Agile principles are derived from Agile values. ASD (Agile Software Development) methods on the other hand consist of values, principle and practices The Agile principles and practices vary greatly across agile methods of software development having evolved from over time to address scalability, global agile development, distributed agile development, Agile-DevOps, agile automation, automated testing and continuous integration (see Dingsøyr and Lassenius, 2016).

Therefore, the important question to investigate is considering that Agile values which when introduced in the Agile Manifesto reflected the fundamental difference and a stark contrast between the two paradigms of software development, have they also diluted overtime or have they endured over the past two decades. In this study we assess if the Agile values are still relevant by conducting a survey with software developers in one of the largest IT consulting firms in the world. We also assess the performance correlates of these 4 values with work outcomes such as efficiency, effectiveness and innovation and affective outcomes such as intrinsic motivation and work exhaustion of team members. Work exhaustion is an important psychosocial outcome as it is associated with absenteeism and job turnover of high-quality employees.

HYPOTHESIS DEVELOPMENT

We expect all four agile values to be positively correlated with affective outcomes, work motivation (higher values of the construct are positive) and work-exhaustion (lower values of the construct are positive) because of their evocative nature and positive emotions associated with as the values on the right side (in bold) represented the yoke of the heavy-weight methods and the left side (in bold) represent the relief. Individuals and interactions over processes and tools (Value 1)

Working software over comprehensive documentation (Value 2)

Customer collaboration over contract negotiation (Value 3)

Responding to change over following a plan (Value 4)

Software Development involves people with different skills and expertise working together to create a new software (Kakakr, 2028cdefg). The key to successful development of software is therefore knowledge sharing between team members. This involves intensive interactions amongst team members to covert embedded knowledge resident in individuals into embodied knowledge in the form of a working software, In knowledge work such as software development the relay race approach in which specialists of one group handover work to the next group of specialists as in plan-driven method is not as effective as the rugby approach of agile methods where the entire team works together throughout the project to reach the finish line in creating innovations (Rigby, Sutherland and Takeuchi (2016). Thus,

Hypothesis 1a: Value 1 is relevant for software development

Hypothesis 1b: Value 1 will positively impact work and affective outcomes

Darwin once said that “A naturalist's life would be a happy one if he had only to observe and never to write.” Programmers love coding but following standards, documenting the code with detailed comments and getting subjected to code inspection is something they generally hate (Kakar, 2020; Kakar and kakar, 2020). However, this value might also create long-term problems. Documentation is used for sharing knowledge between people. With most business softwares needing to be maintained in the long run, the lack of adequate documentation is also responsible for inadequate transition support from development phase to deployment and maintenance phases. The observation that support team members in SD spend a major portion of their time reading source code and the comments they contain are typical of the additional costs that are carried over from the development phase due to lack of detailed documentation. Thus,

Hypothesis 2a: Value 2 is not relevant for software development

Hypothesis 2b: Value 2 will not impact work outcomes but positively impact affective outcomes

Collaborating with the customers throughout the project cycle to meet their evolving needs is more satisfying than tossing over the Systems Requirement Specification (SRS) after requirements gathering from the customer to the development team who then develop code to meet the specifications in isolation of the customer (Kakar, 2018ab). The classic difference between “meeting the specifications” and “fitness for use” is encapsulated in this principle. When employees interact with the beneficiary of their work it motivates them (Grant (2007). Co-creation of software builds trust between supplier and customer and helps break down the traditional barriers between them. ASD can be credited to bring the concept of customer focus rather than product focus for the first time in software development. Therefore, we expect this value to be increasing relevant today and will affect both work and affective outcomes. When team members actively collaborate with customers and understand software from the beneficiary perspective it enhances the significance of the work they do (Parker and Axtell 2001. Further the resulting empathy and positive impact of their work on the beneficiary increases intrinsic motivation and effort persistence (Grant 2007). Thus,

Hypothesis 3a: Value 3 is relevant for software development

Hypothesis 3b: Value 3 will positively impact work and affective outcomes

According to the process control theory (Schwaber, 1995), software development is considered a black box system due to the high of uncertainty in business requirements and rapid changes in technology. The business requirements are not only subject to change due to changes in external environment but also because they are emergent and evolve with better understanding of domain and technologies. The customer should be allowed an opportunity to specify what they actually want from the software rather than what they initially thought they wanted even if it is late in the development cycle (Kakar, 2017abcde). Therefore, applying rigid heavy weight process methodologies and plan-driven approaches are likely to fail for inherently unpredictable and nonrepeatable work. Uncertainty cannot be planned for as unknowns cannot be expected to be understood upfront (Kakar, 2014, Kkar, 2015abc). Therefore, a more fluid approach of planning for short iterations and development of the working product at the end of the iteration. The working product is then evaluated by the customer and changes suggested for the next iteration. Therefore, we expect this value to remain relevant and positively impact work and affective outcomes

Hypothesis 4a: Value 4 is relevant for software development

Hypothesis 4b: Value 4 will positively impact work and affective outcomes

METHOD

To assess the relevance of the Agile Values today we conducted a survey with development team members of 58 recently completed software projects. The developers were employees of a large multinational IT consulting firm with operations across the globe. The 58 projects included software development for 42 companies across 8 countries in North America, Europe and Asia. The type of projects included 22 which were characterized by Project Managers as Waterfall method, 4 V-method, 17 Extreme programming, 10 Scrum, 1 Crystal methodologies, 2 Dynamic Software development method (DSDM) and 2 Feature Driven Development (FDD). The study included responses from 343 developers who answered a survey questionnaire and represented the response from 88% of developers who participated in the 58 development projects. The subjects were of average age 29.3 years and included 181 males and 162 females. The average experience of working in software development teams was 5.9 years.

Measures Used

Subjects responded to all item including the Agile Value question (Do you think Agile Value “Customer collaboration over contract negotiation” is still relevant for Software Development) on a scale with anchors 9=Strongly agree and 1=Strongly Disagree. Team innovation was measured using Tjosvold, Tang, and West (2004) scale. A sample item from team innovation scale is: “The team learned new ways to apply their knowledge of familiar products and technologies to develop new and unusual solutions to familiar, routine problems.” Team Efficiency and Team Effectiveness were measured using the scale

consisting of two sub-scales developed by Hoegl and Gemuenden (2001). The Effectiveness subscale contains items which compare actual versus intended outcomes, while the efficiency subscale contains items related to comparison of intended versus actual inputs. A sample item from team effectiveness subscale is: “All demands of the customers were satisfied.” A sample item from the team efficiency subscale is: “The project was completed within schedule.” Work Exhaustion was measured using the McKnight, Philips and Hardgrave (2009) A sample item from this scale is: “I felt burned out from my work”. The intrinsic Motivation scale was adapted from the Self-Regulation Questionnaire (Ryan and Connell, 1989). A sample item from the scale is “I am glad to have worked on the project.”

Method of Analysis

The reliability and validity of the scales were established using factor analysis. MHMR (Moderated Hierarchical Multiple Regression) analysis was used to determine the correlation between Agile Values and work outcomes. Extraneous variables such as Age, Gender and Experience were controlled for in the analysis.

FINDINGS OF THE STUDY

Factor analysis was conducted to ascertain the reliability and validity of the constructs. The internal reliabilities of the scales were found to be greater than 0.82. From Table 1 we can see that all the four Agile were relevant as the values were all greater than the mid-point of 5 on the 9 point Likert scale Thus, Hypothesis 2a was not supported as we expected this value to be not relevant (<5). However, for team members of Plan-Driven Projects Values 1 and 3 were lower than 5. Value 4 had the highest relevance for team members of both Agile as well as plan-driven projects. Further, hypothesis 1b and 3b were only partially supported as we expected Value 1 and work exhaustion to be positively correlated and Value 1 and innovation to be positively corrected (Table 2). The summary of the results of Hypothesis are presented in Table 3. We can conclude from the findings that all the Agile Values are still relevant today for software development and are overall positively correlated with work and psychological project outcomes.

Measure	Overall Mean	Agile Methods			Plan-Driven Methods		
		Mean	Standard	N	Mean	Standard	N
Value 1	5.53	6.31	0.564	182	4.64	0.597	161
Value2	5.49	5.89	0.781	182	5.04	0.579	161
Value 3	5.66	6.42	0.773	182	4.81	0.741	161
Value 4	6.71	7.37	0.576	182	5.96	0.582	161

Table 1: Descriptive Statistics – Values

Work Outcomes	Motivation	Work Exhaustion	Effectiveness	Efficiency	Innovation
Value 1	.834 **	0.687**	0.441**	-0.501**	0.123
Value2	.782**	0.137	0.150	-0.623**	0.219
Value 3	.821**	0.333*	0.599**	-0.423**	0.475**
Value 4	.925***	0.475**	0.656**	-0.476**	0.524**

* p < .05 ** p < .01 ***p<.001

Table 2. Correlation of Agile Values with Work Outcomes

Value 1	S (H1a)	S (H1b)	S (H1b)	S (H1b)	NS (H1b)	NS (H1b)
Value2	NS (H2a)	S (H2b)	S (H2b)	S(H2b)	S (H2b)	S (H2b)
Value 3	S (H3a)	S (H3b)	S (H3b)	S (H3b)	NS (H3b)	S (H3b)
Value 4	S (H4a)	S (H4b)	S (H4b)	S (H4b)	NS (H4b)	S (H4b)

Table 3. Summary of Hypotheses Supported (S) and Not Supported (NS in bold)

CONCLUSION AND LIMITATIONS

Agile project management was introduced to mitigate the shortcomings of the plan-driven approach in managing uncertainty and change. In heavy weight methods, upfront planning, defined processes, coding standards, inspections and reviews, productivity metrics and statistical quality control was the norm. Managers not only assigned tasks to the team members but also specified how they should be performed (process) and by when (schedule) they should be completed. However, with increasing uncurtaining and change in the internal and external environment the heavy weight methods revealed various shortcomings. Uncertainty cannot be planned for but must be managed. Therefore, the focus shifted to people over processes, intensive customer collaboration iterative development through multiple prototypes, and agility in addressing change. This study found these agile values are still relevant today after the proclamation of the Agile manifesto two decades ago and impacts both work performance and psychological well-being of team members positively. Agile project management principles are now increasing adopted in new product development of physical products and the findings of this study are therefore applicable beyond software development.

However, the findings of the study should be viewed considering the following limitations. The analysis of the results could be performed at broad level of two categories only - Agile and Plan driven paradigms of software development. These two paradigms represent archetypes and are not seen in pure form but in their various hybrid forms. Within each category there are multiple methods each with their own characteristic practices and approaches to software development. The sample size precluded statistical analysis at the level of each method of software development. Future studies may test the validity of the findings for specific methods of software development within these two broad categories

Another limitation of the study is the use of self-report of team members to the variables used in the survey. This raises the issue of common method bias inflating the effect size. However, its effect was mitigated using tested measures and temporal separation in obtaining subject responses on independent and dependent variables (Sharma, Yetton and Crawford, 2009).

REFERENCES

- Dingsøyr, T. & Lassenius, C. (2016). Emerging themes in agile software development: Introduction to the special section on continuous value delivery. *Information and Software Technology*, 77, 56-60.
- Grant, A. M. (2007). Relational job design and the motivation to make a prosocial difference, *Academy of Management Review* (32), pp. 393-417.
- Hoegl, M. and Gemuenden, H. (2001) Teamwork Quality and the Success of Innovative Projects: Theoretical Concept and Empirical Evidence. *Organization Science*, 12(4), 435-449.
- Kakar, A. K. (2020). A Theory of Effectiveness of Agile Software Development. *AMCIS (2020) Virtual*.
- Kakar, A. K. (2014). When form and function combine: Hedonizing business information systems for enhanced ease of use. In *2014 47th Hawaii International Conference on System Sciences* (pp. 432-441). IEEE.
- Kakar, A. K. (2015a). Why do users speak more positively about Mac OS X but are more loyal to Windows 7?. *Computers in Human Behavior*, 44, 166-173.
- Kakar, A. K. (2015b). Investigating the penalty reward calculus of software users and its impact on requirements prioritization. *Information and Software Technology*, 65, 56-68.

- Kakar, A. K. (2015c). Software product features: should we focus on the attractive or the important?. *Journal of Decision Systems*, 24(4), 449-469.
- Kakar, A. K. (2017a). Do reflexive software development teams perform better?. *Business & information systems engineering*, 59(5), 347-359.
- Kakar, A. K. (2017b). Investigating the prevalence and performance correlates of vertical versus shared leadership in emergent software development teams. *Information Systems Management*, 34(2), 172-184.
- Kakar, A. K. (2017c). Assessing self-organization in agile software development teams. *Journal of computer information systems*, 57(3), 208-217.
- Kakar, A. K. S. (2017d). Why do users prefer the hedonic but choose the Utilitarian? Investigating user dilemma of hedonic-utilitarian choice. *INTERNATIONAL JOURNAL OF HUMAN-COMPUTER STUDIES*, 108, 50-61.
- Kakar, A. K. (2017e). Investigating the relationships between the use contexts, user perceived values, and loyalty to a software product. *ACM Transactions on Management Information Systems (TMIS)*, 8(1), 1-23.
- Kakar, A. K., & Kakar, A. (2018a). GTTA 2: AN EXTENSION OF THE GENERAL THEORY OF TECHNOLOGY ADOPTION (GTTA), *Southern Association of Information Systems*.
- Kakar, A. K., & Kakar, A. (2018b). GTTA 3: AN EXTENSION OF THE GENERAL THEORY OF TECHNOLOGY ADOPTION (GTTA), *Southern Association of Information Systems*.
- Kakar, A. K. (2018c). How does the value provided by a software product and users' psychological needs interact to impact user loyalty. *Information and Software Technology*, 97, 135-145.
- Kakar, A., & Kakar, A. K. (2018d). Assessing Shopper's Penalty Reward Calculus in Online versus Instore Shopping. *e-Service Journal*, 10(3), 24-45.
- Kakar, A. K., & Kakar, A. (2018e). IS THE TIME RIPE FOR BRANDING OF SOFTWARE PRODUCTS. *Southern Association of Information Systems*.
- Kakar, A. K. S. (2018f). Engendering cohesive software development teams: Should we focus on interdependence or autonomy?. *International Journal of Human-Computer Studies*, 111, 1-11.
- Kakar, A., & kumar Kakar, A. (2018gb). Is team cohesion a double edged sword for promoting innovation in software development projects?. *Pacific Asia Journal of the Association for Information Systems*, 10(4).
- McKnight, D. H., Phillips, B. and Hardgrave, B. C. (2009). "Which reduces IT turnover intention the most: Workplace characteristics or job characteristics?" *Information & Management* (46:3), pp. 167-174.
- Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). The secret history of agile innovation. Harvard Ryan, R. M. and Connell, J. P. (1989). "Perceived locus of causality and internalization: examining reasons for acting in two domains," *Journal of personality and social psychology* (57:5), pp. 749.
- Schwaber, K. (1995) "Scrum Development Process", presented at *OOPSLA'95 Workshop on Business Object Design and Implementation*. Schwaber, K., and Sutherland, J. (2007). What is Scrum" URL: <http://www.scrumalliance.org/system/resource/file/275/howIsScrum.pdf>, [Stand: 03.03.2008]
- Dingsøyr, T. and Lassenius, C. (2016). Emerging themes in agile software development: Introduction to the special section on continuous value delivery. , 56-60.
- Fowler, M., & Highsmith, J. (2001). The agile manifesto.
- Tjosvold, D., Tang, M. M. and West, M. (2004). Reflexivity for team innovation in China the contribution Group & Organization Management, 29(5), 540-559.