Development of an Agile Workforce, Agilely: A Case Study

Anne Little, PhD
SAIC
Reston, VA
Anne.M.Little@saic.com

ABSTRACT

A 2018 Harvard Business Report which surveyed nearly 1,300 IT and business leaders indicated approximately four-fifths of respondents are "using Agile in some form" to accomplish principal business functions, and three-quarters are "embracing agile ways" (Panditi, 2018). However, how Agile is actually practiced within the organizations is another matter. These same respondents self-reported their implementation consistency was less than 20%. Within the Department of Defense (DoD) and especially noted within cyber warfare operations, the ability to develop software rapidly (usually relying on Agile methods) is critical because "cyber warfare … has become at least as important as physical fighting" (Denning, 2019).

In early 2020, the author led the rapid development of an integrated knowledge management system with software engineers using DevOp and Agile practices. The subject matter experts (SMEs) who would use the system did not have Agile experience, but their engagement with the development team was vital to the success of the end product. Throughout the development, the author noted that nearly every one of the SMEs made claims about "being Agile", and occasionally used Agile terms, but frequently exhibited behavior which was clearly not Agile.

Members of the workforce who have become highly skilled operating in hierarchical or waterfall environments are faced with a challenge if they do not position themselves to make the *transformational* change necessary to become proficient using Agile. Agile and non-Agile practices do not map to each other very well and members who choose an incremental implementation method can become practitioners of "Fake Agile" (Peterson, 2019, Denning 2019b). Additionally, teams who have not been able to bridge operational divides by implementing Agile successfully take on unnecessary technical risk.

This paper presents the methods by which the author used an Agile development project as a vehicle for non-Agile practitioners to develop Agile skills and an Agile mindset. Ways to scale this technique within the broader workforce are also provided.

ABOUT THE AUTHOR

Anne Little, PhD is Vice President for Training Solutions Development with SAIC. She has more than twenty years of training development and delivery experience, and her research interests include motivational strategies within online learning environments. She has leveraged her experience to design training programs for numerous federal clients, including the Department of Defense, Department of Homeland Security, Department of Energy, and the Federal Aviation Administration. Most recently, her research and development portfolio has focused on re-engineering compliance training programs, and knowledge sharing infrastructures. Dr. Little holds a Mathematics degree from Purdue University, and a PhD in Instructional Technology from George Mason University.

Development of an Agile Workface, Agilely: A Case Study

Anne Little, PhD
SAIC
Reston, VA
Anne.M.Little@saic.com

AGILE: THE DRIVING FORCES

Historically, Agile began in the early 2000s as a solution to a problem plaguing the software development industry; software roll-outs failed which caused ripple effects ultimately leading to projects the software was designed to support being cancelled. Root cause analysis of many failures was attributed a substantial amount of developed code not being implemented — not because it had technical defects — but instead, because it was simply not needed. The Agile Manifesto (Beck, et al., 2001) was developed as a response to maximize the time developers spent creating valuable, high-quality software in a quick AND sustainable manner. There was also a human factor element included. Previous operational practices which generated releasable products but included working around the clock and led to high burnout rates were deemed unacceptable. The desired operational state, under Agile, was a team which could operate at productive pace over a long period of time. And, productivity was measured only in terms of working, usable software (or other product which met the value/need).

One tenet of Agile is communication with a clear set of guidelines similar to “rules of engagement”. Small, collaborative teams (usually 4-6 people) which communicate daily help members avoid miscommunications, which can lead to the development of non-usable code. Other meetings, referred to as “ceremonies” include universally understood meeting purposes or objectives, and clear expectations for member participation.

The inherent flexibility of Agile lets teams incorporate the methods which work best for the situation dictated by the project and its current state. One of the most popular methods, Scrum, includes the specific meetings designed to enable a smooth workflow over the course of a specific time — usually two weeks — called a Sprint. Each Sprint contains a clear set of deliverables which are demonstrable so that the product user can provide feedback directly to the development team. This provides the structure for continuous, iterative development and avoids costly re-work.

From a cultural perspective within any workplace, the desired outcomes of implementing Agile (specifically, getting the right products developed as quickly as possible, within acceptable quality standards, and without driving the workforce to the brink of exhaustion and eventually quitting) are very attractive. This has led to organizations deciding “let’s go Agile” before truly understanding what that means. Without careful consideration to the cultural tenets necessary to implement Agile successfully, projects break down, all expectations are missed and everyone questions its value altogether (Suntinger, 2020).

Use and Abuse: Implications of Bad Practice

In addition to Scrum, many other Agile-based techniques including Lean, eXtreme Programming, paired programming and continuous integration, include a flexible aspect. None dictate any one set of practices, which poses a problem for novices who might struggle to understand what is “right”. Additionally, Agile implementation presents a host of unique challenges because it requires a change to the *way* of working, as well as a fundamental adoption of an *Agile mindset* (Denning, 2019a). Therefore, “doing Agile” can be considered both an art and a science, and organizations who adopt the methods (science) without understanding or applying the values and principles (art), can run into problems.

These problems can show up as conflicts between traditional constructs and the need to be flexible. The first is the conflicting way of understanding deadlines. Many organizations decide to adopt Agile because they require the flexibility to adjust their requirements (as they actually see the products being developed in action) to meet their requirements. But, when the organization then wants to see a roadmap of all the expected deliverables over the entire product lifecycle, this simply doesn’t make sense. This is because within an Agile project, the team advances the product such that it meets the requirements (including new requirements as they emerge) by developing and delivering interim functionality, and continuously refining it until it meets the definition of "done" (and no more). Quite simply, under a completely purist ideal (and ultimately unrealistic) Agile operation: 1) there are no deadlines and 2) there is no such thing as scope creep. Unfortunately, these two ideals will likely give your contracting officer and administrators simultaneous heart-attacks.

Alternatively, teams who try to find a compromise by easing their way into Agile practices find none of the successes which are usually associated with either Agile or traditional practices. This is because the inherent baseline structures are at complete odds with each other. Table 1 provides a summary of these values and shows that, while one is not better than the other in all cases, trying to blend the two is actually worse and produces a counterproductive but common practice known as "Fake Agile" (Peterson, 2019, Denning, 2019b).

Table 1: Traditional and Agile Mindset Values: A Merger Leading to Fake Agile

Agile Mindset Values	Traditional Mindset Values
Innovating to deliver steadily more customer value	Focused on delivering shareholder value
Working in small self-organizing teams	Working in teams framed by pre-established rules and roles
Collaborating together as an interactive network	Creating organizations with multiple layers and divisions

The implications of Fake Agile are so worrying that the Department of Defense (DoD) has issued a "Detecting Agile BS" guide (Denning, 2019b). This guide lists warning "flags" or non-practices which, if observed, indicate a program is not using Agile — regardless of whether the team says they are or not. One such flag is "software developers not talking to users". Instead, a hierarchical program structure may direct developers to speak only with a Technical Representative. While this is well within traditional project norms, it poses unnecessary technical risk, but hierarchical structures may blind the program manager to it.

Overall, this need for an Agile BS detector is actually quite disturbing and suggests stakeholders and managers who are responsible for Agile teams have little experience operating within an Agile environment. And, unfortunately, strategies which managers have used previously in hierarchical structures (with consistent success) actually hinder success in an Agile environment. This situation led the author to develop a training engagement which would help a specific group with the following characteristics: 1) at least 10 years' of prior military experience, 2) at least 10 years' experience managing technical or knowledge-work staff, 3) no experience working as a member of an Agile team and 4) some knowledge of Agile terms and/or completion of at least one Agile "course".

To prepare the framework for the program, the author examined some classic organizational change management research paradigms. Specifically, McKinsey consultants' approach (Peters & Waterman Jr., 1982, 2004) which is framed by many interrelated factors, acknowledge that there are many interdependencies associated with a change effort. While this theory seemed like a reasonable approach to introduce Agile, it has known disadvantages. Specifically, because "all the factors are interrelated and interdependent on one another; the failing of one part means failing of all" (Belyh, 2015). In contrast, Kotter's change management theory (1996, 2012), which is one of the most popular and adopted in the world, articulates eight stages which focus on a single principle. Each step must be completed step-by-step and no step can be skipped, making the entire process very time consuming. The author, therefore, believed that this waterfall-like change management model was likely not a great choice to support Agile learning transfer. Instead she focused on a structure which would find a way to examine all of the elements of Agile, but somehow find a way to not create task-overload for the participants.

A CULTURE OF THINKING AND LEARNING TRANSFER

Tishman proposes a culture of thinking as a necessary part of a learning environment. She identifies several driving forces — specifically language, values, expectations, and habits which "work together to express and reinforce the enterprise of good thinking" (Tishman, et al., 1996, p. 2). There are six dimensions to a thinking culture and the author focused on two to frame the treatment and implementation approach towards developing an Agile mindset. These dimensions are: *thinking dispositions* which focus on learner attitudes, values, and habits of mind; and *transfer* which refers to the successful application of knowledge and strategies applied to a context other than the one in which the knowledge was first attained (Tishman, et al, 1996, Thalaheimer & Weber, 2020). These dimensions were viewed as highly relevant to this problem space because successful Agile implementation is underpinned by a common set of values, as well as specific habits and expected use of unique language elements during each team engagement. The dimension of transfer was also viewed as highly relevant, especially since every member of the target audience had participated in some form of Agile training, but no experience in Agile *doing*.

Methods for Evaluating Transfer

Unfortunately, long-held learning-evaluation methods, such as Kirkpatrick (1959), have generated bad practices in measuring learning transfer (Thalheimer, 2018). While the intent of the Kirkpatrick evaluation methods were grounded towards generating positive results regarding behavior on the job, implementation of Kirkpatrick models have led to assumptions which are not backed by research. The following are a few examples of metrics which are frequently used to measure student transfer, and yet, none are supported by research as an adequate way of evaluating learning:

- Completion of a learning event
- Student's level of interest in learning
- Student's level of active participation (and/or attention) in the learning event
- The degree to which the student stated they liked the training event
- Knowledge retention or level to which the learner understands a concept
- The degree to which the learner demonstrates a skill or competency *during the learning event* (assuming that the learner will be able to remember later, which isn't the case)

In their report summarizing current research to "determine factors that could be leveraged by workplace learning professionals to increase transfer success" Thalaheimer & Weber (2020) state that the research base is actually weaker than was expected. While they conclude that practitioners should maintain a "healthy skepticism" about transfer recommendations, there were a few recommendations which are warranted. The key findings from their research which influenced the design of the treatment where:

- Learners who were motivated to apply what they have learned will be more successful in transfer (supported)
- Post-training coaching supports successful learning transfer (*suggested*)
- Providing early opportunities to utilize what has been learned within the workplace will positively impact transfer success (supported)
- A learner can learn poorly during training, but can still be successful in transfer if motivated and engaged in subsequent on-the-job learning (supported)

Therefore, the implemented training engagement included the following elements:

- Timing: the program was released at the point of need; when the learners were motivated to apply what was learned (because they needed to use the skills immediately)
- Integrated self-study with opportunity to observe an actual Agile team demonstrating good models of engagement

Constructivism as a Method to the Art and Science of Learning Agile

Educational experiences based on the educational theory of Constructivism intend to provide learner benefits based on the premise that both "logical analysis of action, and individual experiences generate new knowledge" (Brooks & Brooks, 1993, p. 23). Specifically, a learner's use of particular language combined with the influence of his/her past experience contributes to transfer. Therefore, Constructivist practices are designed to help learners internalize and transform information- as opposed to other methods used to train students to repeat specific procedures or repeat specific chunks of information. Too often, the latter is used as the framework to demonstrate that a student "has learned" where the indicators are results of multiple-choice or short-answer tests. Under a Constructivist paradigm, however, where deep understanding is the goal, versus imitative behavior, it is necessary to provide the opportunity for students to generate a solution, as opposed to repeating one.

Elliot Eisner's view of the American education system — specifically those which steamed from his belief that art should not be studied in absence of other subjects — generated a view of literacy that extends beyond verbal and numerical skills. He believed that art could be studied from productive, critical, and historical aspects, and that it was critical for students to have the language skills necessary to talk about a particular work critically. In his 1994 publication, *Cognition and Curriculum Reconsidered*, Eisner describes problem-based educational practices which the author selected as the methodology to develop the treatment (training plan/approach), specifically because it draws upon an art- and science-based view. It also fundamentally structures learning activities in an integrated manner and demands that learners examine relationships across fields.

Additionally, the author drew upon Vygotsky's research which, although was primarily focused on children, has application beyond childhood development. In fact, Constructivism — the framework for Adult Learning Theory — is traced back to Vygotsky's social learning theories (Eisner, 1994). Specifically, Vygotsky's zone of proximal development (ZPD) separated the concepts of training by using mechanical and mental skills with becoming able to solve a variety of more advanced problems independently (Vygotsky, 1978). Vygotsky's ZPD theories influenced the training design approach of *scaffolding*, which is a process where by an expert provides a model, then steps back to allow the learner to engage, and then engages to offer support when needed.

TREATMENT FRAMEWORK

The literature discussing social learning theory and Constructivism was applied to the training framework to learn Agile based on the following considerations:

- Knowledge transfer (as articulated in research summary completed by Thalheimer, 2020) is a challenge because learners must apply Agile constructs within structures where their prior experience is not relevant nor particularly helpful
- Agile has a series of unique language elements and meetings or "ceremonies"
- Agile teams are expected to work in an integrated manner and adhere to particular social constructs

Therefore, the author developed the training consisting of microlearning, observation of experts modeling Agile processes and opportunity to construct Agile artifacts. The objectives of each of the elements is summarized in Table 2.

Table 2: Summary of Training Objectives

Element	**Objective**
Microlearning assets (short self-paced videos or articles — all open source)	Learners are introduced to a ***single element*** of an Agile ceremony, engagement, or role expectation
Engagement within a virtual Agile Dojo	Learners practice using Agile language within their team
Observation of an existing Agile Team	Learners observe each Agile ceremony, as executed by a mature Agile team working on an existing project
Construction of personal Agile artifacts	Learners apply Agile within their own project, starting shortly after completion of their observations

This complete solution continuously interwove the presentation of new information with observation of the new information in action so the learners were able to appreciate the need for Agile within a relevant context. This provides a kind of "middle ground" for the learner to bridge the gap which occurs when a learner is asked to "learn by doing" but is not yet ready to do so. The author hypothesized that this gap is likely to happen when a learner attempts to learn Agile, because the best use of Agile (by nature) is for inherently complex projects where desired end-state value may be known, but articulate-able requirements are initially not known. Immersing a novice into this environment as a functioning team member puts a large cognitive load on the learner, because the learner has the responsibility of product delivery before they have mastery of the tools designed to help them successfully operate in the environment.

Other problem-based approaches, such as those which introduce a learner to a particular process by immersing the learner into a low-complexity task solely for the benefit of allowing the learner to master the practices (Brooks & Brooks, 1993), have an unfortunate side-effect when it comes to learning Agile. Quite simply, learning Agile through a "manageable problem" to use within a training event, translates into exposing the learners to an example where Agile is applied when there is likely no need for it (i.e., one where requirements are reasonably well-understood). Learners are not in a position to see the value of Agile and may revert to non-Agile processes (and frankly — under these circumstances, where traditional processes work — the reversal is warranted). Because these kinds of training approaches are "wired by" a manageable problem to use within training, it runs the risk of messing up the learning all together, because using Agile isn't appropriate and serves as a poor model. And, unfortunately, immersing a learner into a problem in which Agile is appropriate forces the learner into an overwhelming cognitive experience.

Participants

None of the five participants in this case study reported any experience working on an Agile team, although two reported previous responsibility managing individuals who led Agile teams. The participants were brought together as senior leaders for a project requiring their unique subject matter expertise in multi-domain command and control requirements, development of modeling and simulation environments, and DoD exercise planning.

Two of the participants reported completion of at least two formal Agile courses, the other three reported completion of informal learning events, such as viewing videos, reading articles or other self-study activities.

The participants were picked to engage in the activities as a team, because they were all expected to work on an upcoming project.

The other key players within the study were the members of an existing, experienced Agile team. This team consisted of software developers who were developing a requirements management tool for a government customer. The study participants had access to the development team's Jira board, and were invited to all ceremonies. While the participants were specifically directed to be "silent observers" during each ceremony, the team did offer to engage with each of the participants at the conclusion of each ceremony, in case any of the participants had questions.

Schedule

The author designed the program to be implemented over the course of four weeks or two Sprints. The first three activities were designed to be completed during the first two-week period (Sprint 1) and the last three were designed to be completed during the second two-week period (Sprint 2). During the first Sprint, learners completed microlearning activities which focused on the following topics: 1) Agile Manifesto, Agile vs. Scrum 2) Agile Principles, Kanban and 3) Product Owner, Backlog & Scrum. During the second Sprint, the second set of three activities focused on 1) Scrum Master, User Stories, 2) Backlog Grooming, Sprint Review and 3) Work in Progress, Sprint Retrospective.

Learners could complete the microlearning activities at their own place, but were advised that completing the first three within the first Sprint was optimal because it coincided with the observation of Daily Scrums, Sprint Review and Sprint Planning ceremonies. It is noted that even though the Sprint Review is the final ceremony of a Sprint, the author felt that seeing the "desired end" early in the learning process would be valuable. Additionally, the author felt the Spring Planning would make more sense if the participants had witnessed the review of the previous Sprint. During the second Sprint, the participants completed the final three microlearning activities and observed a Backlog Grooming and a Sprint Retrospective. The integrated schedule is shown in Table 3.

Table 3: Observations by Content Themes

	Activity/Content	**Ceremony/Ceremonies Observed**
(Sprint 1) Activity 1	Agile Manifesto Agile vs. Scrum	Daily Scrums
(Sprint 1) Activity 2	Kanban	Daily Scrums Sprint Review
(Sprint 1) Activity 3	Product Owner Backlog, Scrum	Daily Scrums Sprint Planning

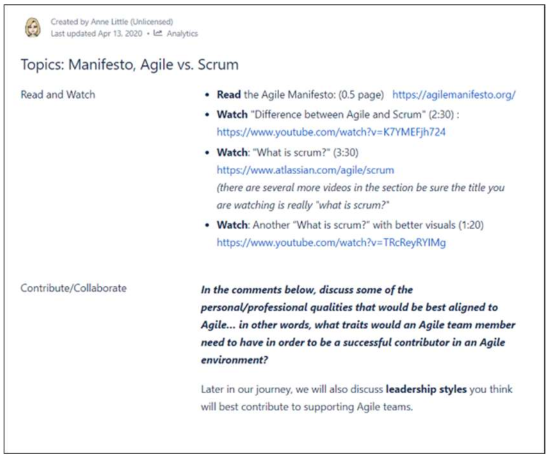
(Sprint 2) Activity 4	Scrum Master	Daily Scrums
(Sprint 2) Activity 5	Backlog grooming	Daily Scrums Backlog Grooming
(Sprint 2) Activity 6	Work in progress Retrospective	Daily Scrums Sprint Retrospective

Format

The author titled the structure of this treatment an “Agile Dojo” and created the training platform within the Atlassian tool, Confluence. This tool is a popular collaboration tool used within Agile teams and it allowed the participants access to the microlearning activities, and provided the ability to comment on them within a single page. The microlearning engagements included short readings and/or videos, and discussion prompts allowed participants to identify elements necessary to develop their own Agile mindset.

As an example, the first microlearning activity (Figure 1) focused on developing mastery of particular Agile terms, in tandem with the observation of Daily Scrums. Participants read a short article and watched short videos.

Each of the six microlearning activities was designed to be completed in approximately 15 minutes.



Created by Anne Little (Unlicensed)
Last updated Apr 13, 2020 • Analytics

Topics: Manifesto, Agile vs. Scrum

Read and Watch

- **Read** the Agile Manifesto: (0.5 page) https://agilemanifesto.org/
- **Watch** "Difference between Agile and Scrum" (2:30) : https://www.youtube.com/watch?v=K7YMEFjh724
- **Watch**: "What is scrum?" (3:30) https://www.atlassian.com/agile/scrum *(there are several more videos in the section be sure the title you are watching is really "what is scrum?"*
- **Watch**: Another "What is scrum?" with better visuals (1:20) https://www.youtube.com/watch?v=TRcReyRYIMg

Contribute/Collaborate

In the comments below, discuss some of the personal/professional qualities that would be best aligned to Agile... in other words, what traits would an Agile team member need to have in order to be a successful contributor in an Agile environment?

Later in our journey, we will also discuss **leadership styles** you think will best contribute to supporting Agile teams.

Figure 1: The first microlearning activity contained within the Agile Dojo

RESULTS

The author reviewed the participant comments which were generated from the prompts within the Dojo pages. The author picked prompts which would encourage the participants to think about mindset, values, characteristics, and personal or professional qualities Agile team members would need to possess in order to be successful. Other prompts asked the participants to identify actions they currently take as part of their regular work habits which they felt were aligned with Agile principles and to identify new actions or habits they felt they would need to further develop. Lastly, some prompts specifically asked the participants about actions they observed during the ceremonies, and why they believed the action occurred or was necessary.

Mindset Comments and Insights

Participants listed the following characteristics as necessary within an Agile team: collaborative, results-oriented, flexible, accountable, open to change and possessing a growth mindset. Each of the participants felt they were already collaborative in nature, but one participant noted that welcoming input from all was something to improve and appreciated the Sprint Retrospective was a good opportunity to facilitate the improvement.

The notion of a Scrum Master (as opposed to a Project Manager) generated interesting comments from the participants. One in particular illustrated the participant’s insight regarding the non-hierarchy of an Agile team. He specifically stated “the team being collectively responsible for the project’s outcome could result in reduced competition between team members” but he also noted that he was concerned about a “team falling into a ‘high school group project’ mentality where inevitably only a few of the team actually do the work.” Another participant responded by stating: “the entire team can see their individual contribution to the whole and can participate in the planning process.” Two of the participants felt that “each individual would be held accountable by their peers”; although they felt accountability was a characteristic an Agile team member should possess on their own.

During the Sprint Retrospective, the participants observed how the experts discussed their previous day’s Sprint Review. They collectively discussed the nature as well as the tone of the customer’s comments. Within a few minutes, the experts had determined several new courses of action to improve their independent technical performance review and document the feedback within Confluence. They also identified the need to build their technical vocabulary and develop a "Project Readme" file.

Agile Activity Comments and Insights

The participants made several comments about user stories and made comparisons to tasks. They specifically commented on why user stories follow the "*as a (role), I want (feature) so that (benefit)*" format. Participants commented that a story provides a "context and understanding about the object"; it "puts things in terms of the expected value of the product to the customer"; and "provides background and motivation behind the request to help the team produce a deliverable or service that addresses the user's concern or desire."

Several comments focused on the nature of interactions with customers in general. They noted that customer involvement was key, and that it would be necessary to provide conditions under which the customer would desire to be involved. They stated "it cannot just be about tools and processes" and that it was up to us to create the conditions under which the "customer would see value in their participation." However, some comments expressed the concern that may emerge if a customer felt there are too many meetings. One of the prompts, therefore, asked the participants to discuss how you might craft a response if your customer made a comment about too many meetings. One of the participants stated that making sure the meeting was well-led and organized was key, and discussed ways to share with the customer how the team would accomplish this. Another commented that the team should communicate to the customer or product owner the need for common product understanding and that their engagement who allow for product development flexibility (without the overhead of completing unnecessary work).

CONCLUSIONS AND FUTURE CONSIDERATIONS

The intent of the case study was to provide a structure for learners to engage in a continuous cycle of micro engagements where the participants would focus on a particular concept (or set of related concepts). After reading or viewing a video providing background information, they would observe the concepts in action as modeled by experts. After observation, the participants shared characteristics they believed were needed to develop an Agile mindset in addition to actions they would need to implement if they were to join an Agile team.

The comments provided by the learners indicated their awareness of mindset values in addition to needed technical expertise. This was shown by comments which stressed methods of interaction and deemphasized tools and processes. Other comments which stated the need for simplicity started additional conversations about the notion of "minimum viable bureaucracy" which was encouraging and showed a desire to not implement structures solely for structure's sake. Upon completion of the program, the participants were assigned to work together on a project, as planned — although much later than expected due to delays associated with the COVID-19 pandemic. They were each able to use what they had learned from the engagements, particularly being able to use the "definition of done" as a method to save time.

Organizations who have several functioning Agile teams who have established practices could easily implement a similar program at a larger scale. It is important that the participants (learners) and the team (experts) have an opportunity beforehand to discuss the ground rules regarding the observations, which the author did in a single 30-minute telecon. The product the software development team in this study was working on was an enhanced version of a product with which the participants were already familiar. This meant the participants had some technical understanding of the product, but they didn't have any development responsibility, so they could focus solely on how the development team operated. It is possible that the structure would still generate positive results for the learners, even if they didn't understand the technical nature of the project they observed. However, the author believes without at least some technical understanding, participants may not fully grasp concepts such as acceptance criteria, or understand the nature of the improvements suggested within the Sprint Retrospective. This could be the focus of an additional study.

In cases where there may not be enough time to implement this program over two full (2-week) Sprints, a future study could adjust the schedule to work with an expert team using 1-week (or possibly shorter) Sprint cycles. Other approaches used within the author's organization have provided coaching support for existing product teams and engaged with them using highly-accelerated (2.5-day) Sprint cycles. While this has generated positive results for teams with members who are experienced, the author is unsure if 2.5 day Sprints would be sufficient time to provide value to novices.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the members of SAIC's Innovation Factory, particularly Fazal Mohammed and Andy Henson. Additional thanks are warranted to the members of the software development team who were gracious enough to allow us to "lurk" within their project, especially Kevin Brensike, Keith Kile, Scott Marshall, Kenta Yoshimura, Nathan Robinson, Cassie Meadows, and Victor Schramm. And, finally, I am most grateful to my dear friend, Mat Winter, for the 5-star top cover to help me get the title right, and "save face".

REFERENCES

Belyh, A. (2015). *Major approaches & models of change management.* Retrieved June 1, 2020, from https://www.cleverism.com/major-approaches-models-of-change-management/.

Cooney, E., Little A. (2015). *Implementation of agile methods within instructional systems design: A case study*. Presented at Interservice/Industry Training Simulation Education Conference.

Creswell, J. W. (2005). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research.* New Jersey: Pearson Education.

Denning, S. (2019a*). Understanding the Agile mindset*. Retrieved May 30, 2020, from https://www.forbes.com/sites/stevedenning/2019/08/13/understanding-the-agile-mindset/#6c2062ff5c17 x

Denning, S. (2019b*). How fake Agile at DoD risks national security*. Retrieved February 26, 2020, from https://www.forbes.com/sites/stevedenning/2019/09/22/how-fake-agile-at-dod-risks-national-security/#11a6b89d8fa8

Dweck, C. (2007). *Mindset: The new psychology of success.* New York: Ballantine.

GO1 Pty, Ltd. (2010). *Your learning & development assumptions: Truth or myth?* Retrieved May 30, 2020, from: https://www.trainingmagnetwork.com/lessons/117905/view?gref=TMNeBook2-5-4-20&no_header=true&referrer_url=https%3A%2F%2Fwww.trainingmagnetwork.com%2Flessons%2F117905%2Foverview%3Fgref%3DTMNeBook2-5-4-20%26slide%3Dstart&slide=1

Kirkpatrick, D. L. (1959). *Techniques for evaluating training programs.* Journal of the American Society of Training Directors, 13(11), 3-9.

Kotter, J. P. (2012). *Leading Change.* Boston: Harvard Business Review.

Lapham, M.A., et al. (2011). *Agile methods: Selected DoD management and acquisition concerns,* Retrieved June 3, 2020, from https://resources.sei.cmu.edu/asset_files/TechnicalNote/2011_004_001_15335.pdf

Northern, C., Mayfield, K., Benito, R., Casagni, M. (2010). *Handbook for implementing agile in Department of Defense information technology acquisition.* The MITRE Corporation, Technical report.

Panditi, S. (2018). *Survey data shows that many companies are still not truly agile*. Retrieved February 26, 2020, from: https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile

Peters, T. J. & Waterman Jr., R. H. (2008). *In search of excellence: Lessons from America's best-run companies.* New York: HarperCollins.

Peterson, O. (2019*). What is fake Agile? Understanding the dark side of agile and how to avoid it.* Retrieved June 18, 2020, from: https://www.process.st/fake-agile/#what%20is%20fake%20agile?

Saba (2020). *Enabling critical skills: A guide for success.* Retrieved March 24, 2020, from http://www.trainingMagNetwork.com

Suntinger, M. (2020). *Stop "going agile".* Retrieved May 30, 2020.from: https://www.atlassian.com/agile/advantage/going-agile

Thalheimer, W. (2018). *The learning-transfer evaluation model: Sending messages to enable learning effectiveness.* Retrieved June 10, 2020, from https://www.worklearning.com/wp-content/uploads/2018/02/Thalheimer-The-Learning-Transfer-Evaluation-Model-Report-for-LTEM-v11.pdf

Thalaheimer, W. & Weber, E. *From training to performance: Transfer research 2020*. Retrieved April 8, 2020, from: http://bit.ly/transfer99WE

Tishman, S., Perkins, D. N., Joy, E. (1995). *The thinking classroom: Learning and teaching in a culture of thinking.* Boston: Allyn & Bacon.

Toner, M. (2020*). The voice of technology*. Retrieved March 24, 2020, from: http://magazine.nvtc.org/2020/03/tech-titan-nazzic-keene/

Vignettes Learning: *Digital transformation must be a priority for learning leaders in 2020.* Retrieved on April 14, 2020, from: https://www.trainingmagnetwork.com/lessons/117870/overview?gref=TMNWP3-4-14-20

Vignettes Learning: *Workflow learning: Why learning must start with work.* Retrieved April 7, 2020, from https://www.youtube.com/watch?v=8uU6ROBnvcI&feature=youtu.be

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.