The Advantages of Managing IT Projects Using Agile / Scrum

Matthew R. McCutcheon

Brandman University

CSCU 498: Capstone Projects

Dr. Walt Bunch

May 2, 2021

Abstract

Traditional project management uses the Waterfall method that follows an in-depth plan established before work starts. Every step is well documented; elaborate Gant charts show the project's timeline and establishes a high level of control to reduce changes once the project begins. The Waterfall method sounds ideal, yet many information technologies (IT) projects fail to meet the three project constraints, scope, cost, and time. The Waterfall method fails to effectively adapt to change and is not suited for effectively managing projects in an industry like IT that faces new developments every day. I aim to investigate the advantages that Agile and Scrum have over the traditional Waterfall method.

Agile approaches and methods are blanket terms used to cover various non-traditional frameworks, methods, and approaches. The Agile approaches share the Agile Manifesto mindset established in 2001 by 17 leaders in software development. The method I used to determine if a project uses Agile is compliance with the Agile Manifesto and its 12 clarifying principles.

Scrum is an Agile development method created by Jeff Sutherland and Ken Schwaber, two of the 17 authors of the Agile Manifesto. Scrum is a popular choice for projects requiring a high level of innovation with a complex scope. Scrum is discussed more in-depth over other Agile frameworks because IT projects are typically more complex and often require innovative teams.

Varying publications by educational organizations and research papers on the implementation of Agile / Scrum are used to determine the advantages of non-traditional project management methodologies. The comparison between the success of Waterfall and Agile / Scrum focuses on the value it brings to the organization and the ability to stay within the project constraints, scope, cost, and schedule.

Agile / Scrum implementation is ideal for IT projects where the requirements often change. The product provides more value to the customer because of continuous stakeholder feedback. Projects managed using Agile / Scrum deliver to the market faster with higher quality at a lower cost compared to projects that use the waterfall method. Organizations that use Agile / Scrum are more efficient and have higher levels of employee satisfaction.

The Advantages of Managing IT Projects Using Agile / Scrum

The digital age revolutionized almost all aspects of business. Companies have continued adapting to technology advances by implementing information technology (IT) projects. Organizations that fail to adapt risk losing business to competitors and the possibility of closing their doors forever. IT projects are necessary to the survival of a business. However, IT and software projects fail at an alarming rate. The issue lies in the complexity and needs for high innovation levels that IT and software projects require.

Traditional project management uses the Waterfall method. This method is easy to understand, well documented, and provides a high level of control. This research paper does not aim to prove the Waterfall method is wrong for every project, but it does identify the disadvantages Waterfall brings to IT projects. This method is effective for projects in many industries, especially construction. However, IT and software projects using Waterfall consistently fail to meet budget and schedule restraints.

This paper identifies the advantages and disadvantages of an alternative technique to manage IT projects, Agile / Scrum. Agile is a blanket term for non-traditional project managing methods, frameworks, and approaches. Scrum is a framework that falls within the Agile mindset. This research paper views studies and educational literature to support the argument that Agile / Scrum has many advantages over the traditional Waterfall method. However, I tried to remain unbiased while conducting research. This paper presents advantages and disadvantages for traditional and non-traditional project management. I aim to determine if there is enough evidence to support the argument that Agile / Scrum is the best way to manage IT projects.

Before determining the advantages of using Agile / Scrum to manage IT projects, this paper will review Waterfall, Agile, and Scrum. First, examining the traditional Waterfall method, its advantages and disadvantages. Then exploring the Agile approach, the Agile Manifesto, and what frameworks fall under the umbrella term. Finally, an in-depth look at Scrum, why it is the Agile framework chosen as the emphasis for this research paper, and why it has become such a popular framework in the IT industry. Followed by reviewing studies and educational material once the reader understands the differences between Waterfall and Agile / Scrum. This section will look at teams that transitioned from Waterfall to Agile / Scrum and compare the likelihood of project success using different methods. Concluding by determining the characteristics of IT projects, how they are different from Projects in other fields, and why Agile / Scrum better serves an IT project's needs.

Project Management Methodologies

Traditional project management uses the Waterfall method. The scope of the project is agreed upon before the project team begins work, and every aspect of the project is well documented. The Waterfall method uses a work breakdown structure (WBS) to create an organized reference of the project scope and all work required to meet the agreed-upon project requirements. The WBS gives a high-level overview of the work to be performed and, according to the PMBOK Guide, serves as the project's road map (2017). The WBS is a great tool when the scope is well defined. However, later in this paper, I will discuss how many IT projects lack a well-defined scope. I mention the WBS here to give an example of how the fundamental elements of the Waterfall method are not always suited for IT projects.

The Waterfall method offers many advantages. The project stages are well defined, the project has a high level of structure, and the concepts behind the Waterfall method are easy to understand. According to Schwalbe, the Waterfall method requires in-depth planning for all project phases; it is well documented and offers a high level of control (2019). If you are a company executive, you might find these traits very appealing. However, Schwalbe gives one more characteristic of the Waterfall method that might change your mind, especially when working on complex IT projects; the Waterfall method slowly adapts to change (2019).

The Waterfall method has a number of disadvantages that do not serve the needs of most IT projects. According to Jeff Sutherland, Waterfall lacks flexibility, changes to the scope are expensive, assumes requirements will not change, and makes it difficult to return to previous stages (2014). When comparing the needs of an IT project like software development to the advantages and disadvantages of the traditional Waterfall method, it becomes apparent that Waterfall lacks the needed flexibility to deliver innovative products within the project restraints. If the scope is fixed and constant changes arise to meet requirements, then we can assume the project will go over budget and schedule. Schwalbe states, “The Waterfall life cycle model has well-defined, linear stages of systems analysis, design, construction, testing, and support. This life cycle model assumes that the requirements will remain stable after they are well defined.” (2019). Assuming the requirements remain stable is one of the most significant disadvantages of using the Waterfall method to manage IT projects.

Agile Project Management

The term Agile is thrown around a lot in organizations today, and its meaning might seem confusing to anyone who has only worked in a Waterfall environment. At its core, Agile is a mindset for managing projects. According to the Agile Practice Guide, in 2001, leaders in the software industry came together and created the Agile Manifesto and 12 Agile principles (2017). This movement started because they identified that the traditional project managing methodology and product development methods were not suited for complex projects as those performed in the software industry. The Agile Manifesto and Agile principles serve as a guide, not a methodology. According to the Agile Practice Guide (2017), there are four values contained within the Agile Manifesto:

1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan.

The four values found in the Agile Manifesto contradict the fundamental elements of the Waterfall method. The Waterfall method is well planned and documented, with contracts used as the foundation of the project's scope. Waterfall also has a high level of control thanks to its many processes. The Agile Practice Guide (2017) also provides the 12 Agile principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity–the art of maximizing the amount of work not done–is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Just like the Agile Values, the Agile Principles do not agree with the traditional Waterfall method.

Many frameworks and methods are now considered to fall under the umbrella term Agile. Some of these frameworks include Kanban, SAFe, XP, and Scrum; the latter will be discussed in-depth later in this paper. These frameworks use the mentality outlined in the Agile Manifesto and Agile Principles while providing structure for teams to follow. Agile frameworks lack the high-level structure found in Waterfall, allowing teams to adapt and become more innovative. The Agile Practice Guide refers to Agile and Kanban as a subset of Lean, stating, "This is because they are named instances of lean thinking that share lean concepts such as: focus on value, small batch sizes, and elimination of waste." (2017, p. 17).

Agile offers many advantages for managing IT Projects. Agile is flexible, transparent, enhances quality, improves customer experience, predictable, and engages the project team and stakeholders (Azanha, Argoud, Camargo, & Antoniolli, 2017). However, these advantages come at a cost; Agile also has disadvantages. Not all projects benefit from incremental delivery; Agile lacks processes, and requires a high level of collaboration (Fitselis, 2008). Despite these disadvantages, Agile has continued to increase in popularity in a variety of industries.

Scrum

As mentioned before, Scrum is an Agile framework that follows the Agile Manifesto and Agile Principles. In fact, the creators of Scrum, Jeff Sutherland, and Ken Schwaber, were a part of the group of software development leaders who wrote the Agile Manifesto and Agile Principles. The Scrum process was created in 1993, predating Agile, as a framework aimed towards software development (Sutherland, 2014). Many concepts that originated in the Scum framework have proven to be so impactful that other frameworks have adopted them.

Scrum places a high level of focus on the Scrum team. The team consists of the Product Owner, Scrum Master, and development team (Schwaber, 2004). The Product Owner is essentially similar to a project manager. They manage the Product Backlog and focus on adding value to the product. The development team is empowered to self-organize and are interdisciplinary. The Scrum Master is an interesting role, especially if you have only used Waterfall. I am a Certified Scrum Master and still find it difficult to explain the Scrum Master's role to people who have never worked with Scrum or Agile. The Scrum Master is in charge of managing the Scrum process while removing obstacles from the team.

Scrum provides several advantages. One of the most significant advantages is its effective use of time and money. This is accomplished due to its sprints, quick development, and quick delivery. Scrum tests the product periodically during sprint reviews offering a high level of visibility and customer feedback. Scrum shows individual contribution throughout the project due to the natural visibility of the framework. Scrum offers increased communication allowing it to adapt to change quickly (Mahnic, & Drnovscek, n.d., p. 12).

The Scrum framework does not measure success in the same way the traditional Waterfall method does. Waterfall considers a project successful if it is completed within the project restraints, the scope, budget, and schedule. Scrum focuses on the value that the deliverables add to the organization. Jeff Sutherland says, "In software development, there is a rule, born out by decades of research, that 80 percent of the value in any piece of software is in 20 percent of the features." (2014). He follows this statement by using an example, a product most everyone has used, Microsoft Word. Think about all the features Microsoft Word has, then think about how many of these features you actually use. Scrum focuses on delivering the features that provide the most value first. Scrum also uses stakeholder and user feedback to implement changes and increase value through the life of the project.

There are a few disadvantages to Scrum. An organization can not simply decide to use Scrum and expect to implement it that day. Scrum requires educating team members on the Scrum process. Scrum also requires the buy-in of the team, organization, and stakeholders. Scrum risks scope creep due to constant changes (Fitsilis, 2008).

How Scrum Works

The Scrum framework is a barebones set of instructions that is adaptable and gives teams the opportunity to add to it. When I was taking the Certified Scrum Master course and exam through the Scrum Alliance, I was a little shocked that the framework did not require three tools that I associated with Scrum. The first tool is my personal preference when tracking and managing work, a Scrum Board. The other two tools were burndown charts and story points. My instructor informed us that these are great tools, but they are not required. Scrum empowers teams, and they can choose what tools work best for them.

The most popular book for traditional project management is the PMBOK Guide; it is 756 pages long. The 2020 Scrum Guide contains only 13 pages and is accessible to the public at scrumguides.com. I mention this to show the simplicity of Scrum compared to the complexity of

Waterfall. However, most people find the Waterfall concepts easier to understand. The reason for their confusion, according to Jeff Sutherland, "It was because of the way people were working, the way most people work. The way we all think work has to be done, because that's the way we were taught to do it." (2014, p. 4-5). Scrum is a different way of working, and the difficulty lies within breaking old ideas.

The Scrum Guide defines Scrum as, “Scrum is a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems.” (Schwaber & Sutherland, 2020, p. 3). The Scrum Guide contains the Scrum theory, values, teams, events, and artifacts. These core elements are the foundation of any highly functional Scrum team.

The Scrum Theory is made up of three main parts, transparency, inspection, and adaptation. Transparency focuses on all work and processes being visible to everyone, all members performing the work, and all parties who receive the work (Schwaber & Sutherland, 2020, p. 3). Inspection is important to Scrum and involves frequent inspections to identify issues throughout the life of the project (Schwaber & Sutherland, 2020, p. 4). Adaptation is fitting for an Agile framework and refers to Scrum's dedication to continually adjusting the project and the processes used (Schwaber & Sutherland, 2020, p. 4).

The Scrum Values are “commitment, focus, openness, respect and courage” (Schwaber & Sutherland, 2020, p. 4). Scrum teams should also follow the Agile Manifesto and the Agile Principles. An argument can be made that Scrum can be used without Agile. However, Scrum was created by two of the authors of the Agile Manifesto. The Scrum values align with the Agile mindset, and we can assume there are additional benefits when an Agile team uses Scrum. The Scrum team is made up of the Product Owner, Scrum Master, and developers (Schwaber &

Sutherland, 2020, p. 5). Developers make up the members that perform the work, but unlike traditional project teams, they are empowered and play a more significant role. Developers plan sprints, adapt the plan daily, instill quality, hold one another accountable, and many other aspects unheard of in traditional non-management positions. (Schwaber & Sutherland, 2020, p. 4). The Product Owner develops and communicates goals, creates the backlog, orders backlog items, and ensures transparency, visibility, and understanding (Schwaber & Sutherland, 2020, p. 6). The final team member is the Scrum Master. This position has been adopted by many other Agile frameworks and is sometimes referred to as an Agile coach. The Scrum Master serves both the Scrum team, Product Owner, and the organization. According to Schwaber and Sutherland:

> The Scrum Master serves the Scrum Team in several ways, including:
>
> ● Coaching the team members in self-management and cross-functionality;
>
> ● Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
>
> ● Causing the removal of impediments to the Scrum Team's progress; and,
>
> ● Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.
>
> The Scrum Master serves the Product Owner in several ways, including:
>
> ● Helping find techniques for effective Product Goal definition and Product Backlog management;
>
> ● Helping the Scrum Team understand the need for clear and concise Product Backlog items;
>
> ● Helping establish empirical product planning for a complex environment; and,
>
> ● Facilitating stakeholder collaboration as requested or needed.

> The Scrum Master serves the organization in several ways, including:
>
> - Leading, training, and coaching the organization in its Scrum adoption;
> - Planning and advising Scrum implementations within the organization;
> - Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
> - Removing barriers between stakeholders and Scrum Teams. (Schwaber & Sutherland, 2020, p. 6-7).

The Scrum Master plays a vital role to a variety of people involved in the project. It is important to remember that the Scrum Master is not a manager; they are a facilitator and servant leader.

Scrum events consist of the sprint, sprint planning, daily Scrum, sprint review, and sprint retrospective (Schwaber & Sutherland, 2020, p. 7-10). A sprint contains all other Scrum events and is no longer than a month. The sprint begins with the sprint planning. The sprint planning involves determining what items from the backlog the team will complete during the sprint and how completing them adds value (Schwaber & Sutherland, 2020, p. 8). The daily Scrum is a concept that many organizations use, often called a daily standup meeting. The daily Scrum is no longer than 15 minutes and gives the Scrum team the opportunity to discuss their progress and make any adjustments (Schwaber & Sutherland, 2020, p. 9). The sprint review examines the outcome of the sprint, determines ways to adapt for future sprints, and the Scrum team presents their progress to the stakeholders (Schwaber & Sutherland, 2020, p. 9). The sprint retrospective is when the team evaluates processes and determines how they can increase quality and efficiency (Schwaber & Sutherland, 2020, p. 10).

There are three Scrum artifacts, the Product Backlog, Sprint Backlog, and Increment (Schwaber & Sutherland, 2020, p. 10-11). The Product Backlog is a list of needs to improve the product (Schwaber & Sutherland, 2020, p. 10). Items are taken from the Product Backlog and added to the Sprint Backlog to identify what the team will complete during the sprint (Schwaber & Sutherland, 2020, p. 11). An Increment is a predetermined milestone (Schwaber & Sutherland, 2020, p. 11).

The Scrum Guide gives teams only what they need and is intended to be added to by the team to meet their needs and the needs of the project. Schwaber and Sutherland consider the framework intentionally incomplete, also stating, "Various processes, techniques and methods can be employed within the framework." (Schwaber & Sutherland, 2020, p. 3). This prevents teams from being restrained by the rules of an overly complex framework or methodology like Waterfall, where the rules can often hinder the project.

Why Scrum Works

Scrum works because it relinquishes control and empowers teams. This concept defies human nature. As a project becomes more complex, executives and managers tend to increase their control over the projects. The thought of allowing individual team members to make decisions for the organization is scary enough to keep CEOs awake at night. However, no matter how hard they try, increased complexity comes with less control. According to Schwaber:

> As complexity increases, central control and dispatch systems breakdown. Some might try valiantly to make control systems work by applying more rigor, and indeed that works for a while. But the people who prevail are those who figure out how to change to a system of independent agents operating under an appropriate set of rules. (2004).

IT projects are complex, and the control that traditional project management methods impose will break down due to increased complexity.

The shortened feedback loop that Scrum provides also contributes to the framework's success (Schwaber, 2004). The Scrum team periodically meets with stakeholders to discuss the product during development. A software developer might discuss the product and features with the customer at the end of the project in a traditional setting, or they might not have any one-on-one communication with them. In fact, that developer would be breaking the rules of the Waterfall method if they made any changes to the product without notifying the project manager, submitting a change request, waiting for the change request to be approved, updating the scope statement, contracts, WBS, etc.

Scrum approaches sprints as a cycle of learning. Schwaber says, “If we already know everything and have nothing to discover, perhaps we don’t need to use Scrum.” (2004). I completely agree with this statement. If we look at a project like building a house, we can assume that the contractors, plumbers, electricians, carpenters, etc., already know their trades. The team knows how to build a house. This team does not need to use Scrum. Now imagine taking the same team and having them build a boathouse. The team now has to use their skills to create something that will require them to learn and adapt as they go—the complexity of IT projects and other projects requiring innovation benefit from Scrum.

Scrum prioritizes the most valuable deliverables and delivers them first. Scrum is value-driven and uses the framework's short feedback loop to revaluate and re-prioritize continually. A feature might seem relatively crucial at the beginning of a project, but as the team learns and receives feedback, they might determine it does not provide the amount of value as they previously thought. The item is then placed lower on the backlog. Likewise, the same thing could happen with a backlog item whose value is determined to be more important than initially thought, and the team would place it higher on the backlog.

Scrum allows many brains to work together to solve a problem. The ideas of the team are shared on a daily basis giving everyone equal input through increased communication. Perspective at every level of an organization has value. Sometimes situations that could cause major issues are not visible to executives, but blatantly obvious to entry-level employees. Schwaber states, "We know that when things go wrong, there are people around who knew there was a problem, but somehow their ideas were overlooked." (2004). Scrum empowers teams to manage themselves, as a group, and as individuals. Small teams encourage participation, giving every member a feeling of control (Schwaber, 2004).

Why Agile / Scrum is Better for Managing IT Projects

The reason Agile / Scrum is the best option for managing IT projects lies in their origins. Both Agile and Scrum were developed by people who saw software development projects using the traditional Waterfall method constantly go over budget, over schedule, fail to meet scope, and deliver a product that added little to no value to the organization. These individuals knew there had to be a better way. They dedicated themselves to developing a mindset and framework to address the needs of projects in a growing industry with complex projects that require higher levels of innovation. According to Sutherland, “The reason this framework works is simple, I looked at how people actually work, rather than how they say they work.” (2014, p. 8). The traditional Waterfall method resists changes. The creators of Agile and Scrum knew that IT projects required changes to succeed and empowered teams to implement changes throughout the life of the project.

The IT industry sees constant changes at a rate that is not seen in any other industry. When the ancient Egyptians created the Great Pyramids of Giza, what outside forces would impact their project and require changes? Maybe the weather, an earthquake, or some other black swan event. The Egyptians already spent generations creating pyramids and learning from their mistakes. This time they were doing the same project, just bigger. The project requirements were stable. Unlike the construction of ancient monoliths, today's IT industries see changes every day that can impact a project altering its requirements. Software and firmware updates occur all the time. Changes that impact compatibility arise. The IT industry demands a flexible mindset and framework to adapt to all these changes. Agile and Scrum offer teams the flexibility to adapt to these changes. Most IT projects have never been done before. Sure, similar projects might exist, but due to the complexity of the projects, it is uncommon to perform the same project using all the same software, hardware, coding languages, etc. Scrum's sprints and feedback loops allow the team to learn, adapt, and innovate.

IT projects face an increased need for changes based upon the nature of the work performed. According to the Agile Practice Guide (2017):

> Teams can plan and manage projects with clear and stable requirements and clear technical challenges with little difficulty. However, as the uncertainty in the project increases, the likelihood of changes, wasted work, and rework also increases, which is costly and time consuming. (p. 14).

The Agile Practice Guide (2017) then gives approaches that successful teams use to adapt to changes quickly:

- Very short feedback loops,
- Frequent adaptation of processes,
- Reprioritization,
- Regularly updated plans, and
- Frequent delivery. (p. 15)

This list aligns with the Scrum framework. Scrum uses two to four-week sprints, adapts processes at the end of every sprint, re-prioritizes, and updates the plan utilizing the backlog, and delivers at the end of every sprint.

Agile / Scrum breaks up complex projects into sprints. This advantage means teams have a working product every two to four weeks and theoretically can ship the product at any time. This also allows organizations to distribute products incrementally. The first increment contains the features with the most value. This approach benefits software development, networking, electronics, etc. In any industry, beating your competitors to the market offers a considerable advantage. Agile / Scrum gives organizations in the IT industry a competitive edge allowing them to get their product to the customer first while providing additional functions over time.

Stakeholders do not always know what they want. They might know what problems they have or what areas they would like to improve, but they might not know how to accomplish their goal. This is a huge issue when using the Waterfall method that requires the scope of the project to be established before work begins. Agile / Scrum thrives in this kind of environment. Agile / Scrum receives stakeholder feedback during the project and makes adjustments based on the feedback given. The project team does not tell the customer what they want; instead, they work together to find the solution.

When you think of IT projects, one of the first words that often comes to mind is innovation. Many IT projects require creating something new, something that has never been done before. This requires innovation. Agile / Scrum creates an environment that promotes innovation. Teams are empowered to be creative and work together to solve complex problems. High levels of communication allow teams to discuss ideas, receive feedback, and cooperate in ways that are not seen in a project using the Waterfall method. The communication and innovation advantages mean that teams can solve complex problems faster.

There is a term used in the Army that has never made sense to me, a force multiplier. This means that there is a problem, and to solve it, we will throw more Soldiers at it to show we are trying to fix the problem. From my experience, this never helps. Scrum focuses on small teams of six to ten. Selecting the right members for the team depends on the skills the project required and the skills the individuals possess. Scrum goes a step further and empowers these teams. The Scrum master removes obstacles and prevents outside interference. IT projects often require several different skill sets. Scrum provides teams with the required skill sets while also keeping the teams small enough to keep communication levels high and remain innovative.

Agile / Scrum performs periodic testing at the end of every sprint. This is extremely important when working on complex projects. It is easier to fix issues with a product through the development process than waiting until the end only to find out there are issues. This advantage serves all IT projects due to their complex nature.

One of the biggest downfalls I find with the Waterfall method is the lack of user input. When developing a product, it only makes sense that the people using it should be considered. Scrum not only looks for stakeholder feedback at the end of every sprint, but it also gets user input (Sutherland, 2014). This leads to more efficient and desirable products for the end-user who will be using it every day.

Agile / Scrum Research and Findings

This paper has mainly focused on educational material and will now shift focus from theory to the findings of real-world research. It is impossible to create two identical projects with identical teams, allowing us to directly compare Waterfall and Agile / Scrum under the same conditions. However, there is still an abundance of research and interviews to examine and determine if Agile / Scrum is ideal for managing IT projects.

A case study was performed of a pharmaceutical company's IT project in Brazil that used Agile / Scrum. Its goal was to compare Agile project management to Waterfall. The study found many benefits to using Agile / Scrum, including "increased motivation and staff satisfaction, better control of requirements and especially higher quality of the delivered system, generating added value to the organization." (Azanha, Argoud, Camargo, & Antoniolli, 2017). The research team also found benefits in delivery, stating, "Additionally, the project allowed the use of features from the first month of the application deployed, enabling a 75 percent reduction in development time, compared to traditional methods." (Azanha, Argoud, Camargo, & Antoniolli, 2017). The traditional methods refer to the Waterfall method that historically has been the standard in project management. Azanha, Argoud, Camargo, & Antoniolli provide findings that would encourage even the most skeptical manager to try Agile / Scrum, stating, "The software development time was four months, 30 percent of what would be the total if the traditional methodology was adopted. Based on the results, the agile framework, especially the Scrum, proved to be a viable option as a project management approach." (2017).

Viljan Mahnic and Slavko Drovscek from the Faculty of Computer and Information Science at the University of Ljubljana wrote a paper describing their experience using Scrum for software project management. The project was for the university's student records information system. The paper gives an elaborate overview of Scrum, followed by how it was implemented into their project. The conclusion provides the advantages the team found using Scrum. Mahnic and Drovscek found communication benefits stating, "The use of Scrum improved the communication among them and maximized co-operation" (n.d., p. 6). The team also saw benefits to motivation and talent management, stating, "It also increased their motivation and responsibility for the success of the project. On the other hand, it gave them freedom to maximally exploit their talent and knowledge during each Sprint." (Drovscek & Mahnic, n.d., p. 6). Empowerment gives teams the freedom to self-organize. The team determined who would do what based on personal preferences and skills (Drovscek & Mahnic, n.d., p. 6). After the project, the team felt good about their performance as individuals and as a whole (Drovscek & Mahnic, n.d., p. 6).

Drovscek and Mahnic also present the Product Owner and Scrum Master's findings. They found the framework transparent, manageable, and were happy with the ability to rapidly detect and eliminate issues. In their conclusion, Drovscek and Mahnic state, "All impediments were immediately detected during Daily Scrum meetings and removed as soon as possible." (Drovscek & Mahnic, n.d., p. 6).

In 2019 Enric Hidalgo published an article that presented the finding of interviews conducted with 17 researchers regarding the adoption of Agile project management. Hildalgo found that teams that implemented Scrum and the Agile mindset did so due to "key features of the scrum framework and agile methodology such as flexibility, autonomy and self-organization" (Hidalgo, 2019).

Hidalgo's article focuses on the implementation of Scrum at the UK-based Centre for the Evaluation of Complexity Across the Nexus (CECAN). As you read the article, it seems that the implementation of Scrum and Agile fails. However, when we take a closer look, the organization says they were using the Scrum framework, but in reality, they did not. This is a fact that the author fails to mention or is not educated on Scrum enough to realize. The requirements of Scrum outlined in the How Scrum Works section of this paper are the requirements for the framework. Additional tools and processes can be added to meet the needs of the team and organization, but nothing outlined in the Scrum guide can be taken away. Hidalgo mentions several things within the article that make it obvious that Scrum was not properly implemented.

CECAN did not properly implement an important Scrum role, the Scrum Master. Hidalgo states, “This key role was performed by CECAN researchers instead of professional Scrum Masters, and was focused on coordination, facilitating connections and providing guidelines for specific case studies." (2019). For the framework to be considered Scrum, the Scrum Master is not also a developer, or researcher, or any other role, they are only a Scrum Master, and they only focus on performing the duties of a Scrum Master. Additionally, the article gives the impression that the individuals were not qualified for the Scrum Master role.

CECAN neglected the proper use of Scrum events, meaning the framework they used can not be considered Scrum. Hidalgo states, "With respect to the adoption of Scrum methods at CECAN, the concept of sprints was less explicit or used among participants. For example, the practice of establishing regular “standup” meetings, or retrospective meetings at the end of each sprint period was not routinely followed” (2019).

The framework used by CECAN also lacked the role of the Product Owner. A CECAN manager managed a Trello board for several projects, but the responsibilities associated with a Scrum Product Owner were shared between the team members and the so-called Scrum Master. The Scrum Master was the link between the team and the stakeholders, while the CECAN manager was responsible for the backlog (Hidalgo, 2019).

I mention Hidalgo's article in this paper to provide an example of how an organization can say they use Scrum, but this is not always true. Sadly, Organizations like CECAN who do not properly implement the Scrum framework, will report that Scrum and Agile did not work for their organization. The organization either lacked education on the Scrum framework, or managers were reluctant to buy into a new way of working completely. It is obvious that the team was not properly trained and lacked the value that professional Scrum Masters bring to an organization.

The Benefits of Agile / Scrum is Not Limited to IT Projects

Agile / Scrum's focus on software development might make it seem like they are options limited to IT projects. However, this is not the case. They might have originated as responses to the needs faced by software development, but their benefits transition into other industries. Schwalbe supports this by stating, "Remember that Agile methods can be used on all types of projects, not just software development, and several projects use a hybrid approach where some deliverables are created using more traditional approaches." (2019, p. 363). Different Agile frameworks, like Scrum, can be implemented or combined with other frameworks regardless of the project's industry. Sutherland states, "Scrum has its origins in the world of software development. Now it's sweeping through myriad other places where work gets done." (2014, p. 203). Sutherland then goes on to express that even he, the co-creator of Scrum, is surprised at the number of organizations in a variety of industries that have adopted Scrum (Sutherland, 2014).

Thomas Gustavsson from Karlstand University performed a literature review analyzing the use of Agile in non-software development industries. Gustavsson states, "The popularity of agile project management methods are growing and research shows an increasing amount of successful projects due to the transition into agile project management" (2016, p. 3). Gustavsson analyzed 21 case studies published between 2006 and 2015, finding 17 reported benefits of using Agile in non-software-related projects (2016, p. 6-7). The benefit mentioned the most, found in 11 articles, was increased team collaboration (Gustavsson, 2016, p. 7). This is not surprising based on the emphasis placed on frequent feedback loops and empowered teams. The second most frequently mentioned benefit was increased customer interaction, mentioned in nine articles (Gustavsson, 2016, p. 7). This is a result of the use of sprints and continuous feedback from the stakeholders and users.

The third most mentioned benefit in Gustavsson's list is one that should catch the eye of any organization's executives and managers. Mentioned in eight of the articles, increased productivity and speed (Gustavsson, 2016, p. 7). These same members of an organization would also be interested in the seventh most mentioned benefit, increased quality, and other benefits further down the list like decreased customer complaints and improved resource allocation (Gustavsson, 2016, p. 7).

Other benefits include increased flexibility and change, understanding goals and requirements, transparency and visibility, added value, knowledge sharing, cross-organizational collaboration, increased focus, removal of impediments, individual autonomy, motivation, and a clear understanding of progress (Gustavsson, 2016, p. 7). These benefits define an organization with positive work culture, happy employees, and a business that attracts talented individuals.

Gustavsson also found some reported challenges related to implementing Agile in non-software-related projects and organizations; however, the frequency of the challenges was much lower than the benefits mentioned. The number one reported challenge did not stem from Agile or the implemented framework. Gustavsson stated the most commonly reported challenges were changing the mindset of individuals to allow change and receiving buy-in from managers (Gustavsson, 2016, p. 8). This means that the people within the organization stood in the way of obtaining the benefits Agile offers their organization. To address challenges with individual mindset and management buy-in, everyone within the organization needs to be educated on

Agile, the framework the organization chooses to use, and the benefits of the changes taking place during the transition. It is human nature to resist change. Most employees and managers will resist changes they don't understand. However, if they are shown that the changes will benefit them in the long term, they are more likely to accept the changes. These challenges also prove the importance of Scrum Masters and Agile coaches whose job is to coach teams and ensure Agile and Scrum practices are followed.

Gustavsson makes an interesting observation when comparing the Agile values found in the Agile Manifesto to the benefits most commonly mentioned in the 21 studies performed on non-software-related projects. Gustavsson states in his conclusion, "Also the highest amount of benefits corresponds to the first value of the Agile manifesto "Individuals and interactions over processes and tools". These findings are interesting for further research since they imply that the first value of the agile manifesto could have the largest impact on benefits in agile applications in non-software development contexts." (2016, p. 9). Based on this statement, it appears that process-heavy frameworks and methodologies, like the Waterfall method, hinder software development and negatively impact projects in other industries.

Conclusion

Advances in technology have completely changed how businesses operate. Organizations must adapt to remain relevant and competitive or risk closing their doors forever. These changes have increased the need for IT projects. Many IT projects fail to finish within the project's restraints or are never completed. Organizations face many issues using traditional project managing methods due to the complexity and innovation that IT projects require. Organizations like the level of control and familiarity offered by the waterfall method. However, Agile / Scrum provides several advantages to managing IT projects.

Agile is often used as a blanket term that includes several frameworks for project and product managing. Agile is deeply rooted in a mindset covered in the Agile Manifesto and Agile principles. Scrum is an Agile framework that focuses on empowering small teams, sprints, and value. Together, Agile / Scrum offers a more effective way to managing projects that meet the needs of an organization by working with the stakeholders and users during development, collecting their feedback, and catering the product to their needs.

Agile / Scrum offers many advantages and is especially suited for software development. This does not mean that Agile / Scrum is the best way to manage all projects. Agile / Scrum has many disadvantages. The most significant drawback is that not all projects benefit from incremental delivery. Agile suffers from poorly defined processes and is dependent upon collaboration. Scrum requires experienced team members, risks scope creep, and involves education and buy-in.

Agile / Scrum is better suited for managing most IT projects. Agile/ Scrum was developed by leaders in software development to address many of the issues IT projects face when using the traditional waterfall method. The flexibility Agile / Scrum offers allows projects to adapt to change, perform frequent tests, break down complex problems, adjust projects based on stakeholder feedback, and promote innovation. These advantages enable the project to adapt over time, leading to an end product whose success is determined by the value it adds. This is much different than the Waterfall method, where success is determined by predetermined project constraints and change processes that are slow and costly. Organizations and IT project managers still using the Waterfall method should strongly consider transitioning to Agile / Scrum due to the advantages it provides to IT project management.

References

Agile Practice Guide. (2017). Project Management Institute.

Azanha, A., Argoud, A.R.T.T., Camargo Junior, J.B.d. & Antoniolli, P.D. (2017). Agile project management with Scrum: A case study of a Brazilian pharmaceutical company IT project. *International Journal of Managing Projects in Business*, 10(1), 121-142. https://doi.org/10.1108/IJMPB-06-2016-0054

Fitsilis, P. (2008). Comparing PMBOK and agile project management software development processes. *Advances in computer and information sciences and engineering,* 378-383. Springer. Dordrecht. https://doi.org/10.1007/978-1-4020-8741-7_68

Gustavsson, T. (2016). Benefits of Agile Project Management in a Non-Software Development Context: A Literature Review. *Project management development – practice and perspectives: Fifth international scientific conference on project management in the Baltic countries, conference proceedings*, 114–124. http://urn.kb.se/resolve?urn=urn:nbn:se:kau:diva-70311

Hidalgo, E. S. (2019). Adapting the Scrum framework for agile project management in science: Case study of a distributed research initiative*. Helion,* 5(3). https://www.sciencedirect.com/science/article/pii/S2405844018340635

Mahnic, V., & Drnovscek, S. (n.d.). *Agile software project management with Scrum*. University of Ljubljana, Faculty of Computer and Information Science. https://www.researchgate.net/profile/Viljan-Mahnic/publication/228967959_Agile_Software_Project_Management_with_Scrum/links/09e4150644170a03ea000000/Agile-Software-Project-Management-with-Scrum.pdf

Pmbok guide: A guide to the project management body of knowledge (6th ed.). (2017). Project Management Institute.

Schwalbe, K., (2019). *Information technology project management (9th ed.).* Cengage.

Schwaber, K. (2004). *Project management with Scrum*. Microsoft. https://books.google.ae/books?hl=en&lr=&id=6pZCAwAAQBAJ&oi=fnd&pg=PT9&dq=agile+scrum+project+management&ots=kcpTRZ6qgW&sig=5LP6yKryHBtqZ1bS0a3VQVnbxSY&redir_esc=y#v=onepage&q=agile%20scrum%20project%20management&f=false

Schwaber, K., & Sutherland, J. (2020). *The scrum guide.* Scrum Guides. file:///C:/Users/Matt/Desktop/CSM/2020-Scrum-Guide-US.pdf

Sutherland J., (2014). *Scrum: The art of doing twice the work in half the time.* Currency New York.