

THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMMUNICATION ARTS AND SCIENCES

Understanding Agile Project Management: Implementations & Case Study

JACOB IWINSKI
SUMMER 2022

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degrees
in Communication Arts and Sciences & Economics
with honors in Communication Arts and Sciences

Reviewed and approved* by the following:

Jeremy Engels
Professor of Communication Arts and Sciences
Thesis Supervisor

Mary High
Associate Teaching Professor of Communication Arts and Sciences
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

Agile is a term that covers a modern approach to project management that stresses collaborative problem solving, self-organization, and the promotion of cross-functional teams. While the roots of Agile can be traced back to the 60s, it did not see mainstream adoption until the software industry boom over the past 30 years. Since then, Agile project management has rapidly expanded to every industry and market. Because of the inherently ambiguous form and inconsistent implementation of Agile, people often become confused as to its true definition and purpose. This paper has three primary goals. The first goal is to define and contextualize Agile within its history. Second, to review current communication research on Agile. And third, to interview practitioners of Agile in the private sector to contextualize the lived experience of workers and management. While limited in scope, through these three goals a broad overview and general perspective on Agile may be obtained.

TABLE OF CONTENTS

LIST OF FIGURES	iii
LIST OF TABLES	iv
ACKNOWLEDGEMENTS	v
Chapter 1 Introduction	1
The Only Thing Constant is Change.....	1
Paradigm Shift: Waterfall to Agile	3
The Agile Manifesto	5
Chapter 2 Agile Implementations	7
Scrum	8
Kanban	10
Chapter 3 Communications Literature	12
Chapter 4 Analysis	13
Mix & Match.....	13
Purpose of Waterfall	14
Flexibility & Robust Teams.....	14
Democratization & Empowerment	14
Chapter 5 Methodology	15
Chapter 6 Interviews	17
Chapter 7 Discussion	21
Chapter 8 Conclusion.....	22
Appendix A INTERVIEW QUESTIONS	23
Appendix B INTERVIEW TRANSCRIPT – FAANG DEVELOPER	25
Appendix C INTERVIEW TRANSCRIPT – FAANG PROJECT MANAGER, SCRUMMASTER	38

LIST OF FIGURES

Figure 1. Relationship between Lean, Agile, and Kanban.	4
Figure 2. The Agile mindset, values, and principles.	5
Figure 3. General Idea of the Scrum Process	9
Figure 4. A representation of a Kanban Board	10

ACKNOWLEDGEMENTS

The author would like to thank John Gastil (Penn State University), Mary High (Penn State University), Jeremy Engels (Penn State University), and all my friends and family for their time, dedication, and unwavering support throughout the completion of this thesis.

Chapter 1

Introduction

Agile is a mindset. It is a perspective that can be used by an entity for decision making on the smallest to largest scale. It is broad and encompassing while being targeted and focused. It can be combinatory, constantly reiterated and improved upon, and tailored to fit specific use cases. It follows then, for these reasons, that many people struggle to grasp and understand Agile. From inconsistent definitions to improper training, which is often supplemented by lackluster managerial implantation, many workers find themselves working in an “Agile Environment” without possibly being able to define what an “Agile Environment” is. The transition to the “Agile Environment” has been one of the most important shifts to ever occur in project management. It has permeated every industry, at every level of the managerial chain, and wagers to represent a permanent shift. This chapter provides a description of the forces that have guided this change, explain the project management history that forms the foundation for Agile, and develop a robust definition for Agile and summarize its current effect on global workplace organization.

The Only Thing Constant is Change

In the 21st-century, technological, sociopolitical, and environmental factors have placed the modern economy in a constant state of rapid change. The pace of this change even seems to be accelerating, and it has dire consequences for businesses that are stagnant (Neill, 2018; Penning & Bain, 2018). Understanding how to adapt to this change is the only hope of survival

for many organizations which are already in the middle of, or are undergoing, transformational change in the face of one or more of these factors (KPMG International, 2016, 2019).

For companies that successfully adapt to the new world, the rewards can be immense, and they can be just as dire for those that fail. One of the best metrics for measuring power and change in the corporate world is the stock market. The Standard & Poor's 500 index, or S&P 500, is a market capitalization index that includes 500 of the largest publicly traded U.S. companies. In 1964, a given company would have stayed on the S&P 500 for an average of 33 years (Anthony et al., 2018). In 2016, a given company had stayed on the list for an average of only 24 years. By 2027, this corporate tenure is expected to reach 12 years (Innosight, 2019). This shorted lifespan is indicative of the disruptive factors taking hold in the world and is a clear warning sign to stagnant companies without plans for broad scale institutional change. Merely maintaining the status quo has become insufficient; to survive, innovation becomes necessity.

In response to this new paradigm, CEOs and their staff across the world have come up with a single word that they will often use as their stalwart means of defense against this monsoon of change – “Agile” (or “Agility”). When recently surveyed, over two-thirds of chief executive officers believed that Agility is the new currency of business. They also believe by extension that, if they fail to adapt, they and their business will become irrelevant. (KPMG International, 2019).

The Agile mindset is comprised of the Agile principles, values, and best practices; it has been implemented via several more specific methods or approaches such as those named Scrum, Kanban, and XP (Ragas & Ragas, 2021). The general mindset was developed in the US software industry (Silicon Valley) and built off Lean manufacturing principles developed by Honda and Toyota in 1980s Japan (Carvalho & Mello, 2011). The Agile mindset represents a paradigm shift

in corporate America, but before its key differences can be explained, it is necessary to define what had existed before.

Paradigm Shift: Waterfall to Agile

Classic organizational management is often represented through the term “command-and-control,” as it is modeled after traditional military hierarchies. Key aspects include rigid chains of command and regimented specialization of tasks through silos. An organization of this type is often characterized as a machine: steady, slow, and heavy, but with focused intent (Ragas & Ragas, 2021). Absent of forces which pressure to adapt and overcome, it is easy to understand why organizations have traditionally prioritized stability in their project management principles. As change and risk are inherently intertwined, it is logical to minimize risk whenever and wherever possible if change is not necessary for survival. When implementing project management strategies under the old system, the necessary steps for any project would be laid out sequentially in advance. A project plan with sequential steps guided teams throughout the duration of the project. Everyone on a team would know upfront what their roles, responsibilities, tasks, and deadlines would be from the beginning (Hughey, 2009). This style of project management is now known as “Waterfall,” originally defined by Winston W. Royce in 1970. It is like how the steps and rocks of a Waterfall are immovable and placed ahead of time. The project trickles and flows down each step until it reaches its logical end state in a predictable fashion (Royce, 1970). While every step might change slightly in execution, the goal was always to forecast and adjust for these contingencies within the plan. For the majority of project management history, this process was adequate for almost all projects and procedures.

In the globalized 21st-century world, this equation no longer holds true. Adaptability is now essential to organizational survival. The first major innovation in the project management space was “Lean” manufacturing. Inspired by the Toyota Production System (TPS) this style of product management uses “value stream mapping” to separate all aspects of a project into two separate piles. Activities that provide value to the customer and activities that are waste and should be discarded (Liker, 2004). Ideas surrounding TPS and value stream eventually merged into Lean thinking and Kanban, a way to organize knowledge work that was originally inspired by Lean (Ragas & Ragas, 2021). Figure 1 shows the relationship between modern iterations of Agile project management with reference to Lean.

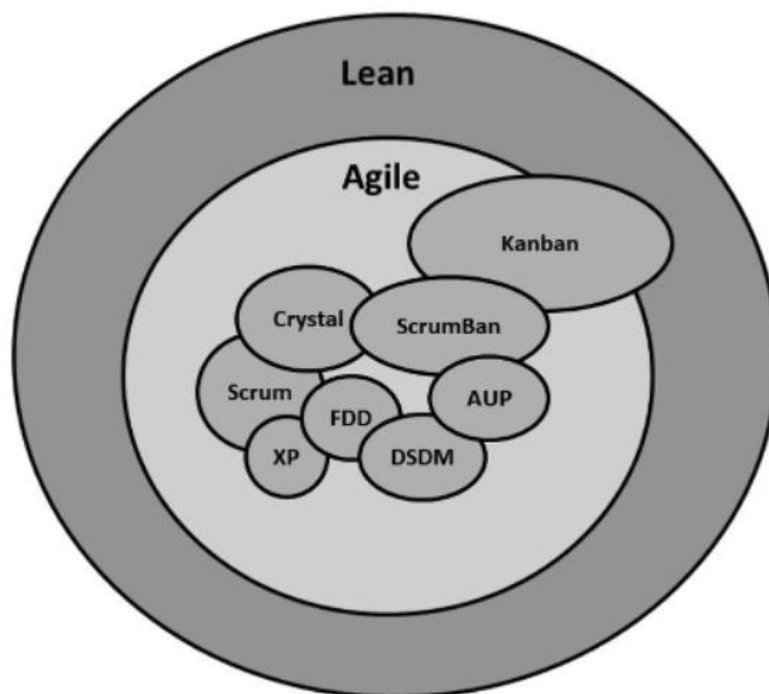


Figure 1. Relationship between Lean, Agile, and Kanban. Source: (Ragas & Ragas, 2021)

Lean and its associated Toyota Production System represent the core of what Agile is built from and shows the way in which project management has evolved to be continuous and iterative. Agile is not an entirely new idea but rather an evolution of existing best practices.

The Agile Manifesto

The Agile Manifesto is a collection of four values and 12 principles posted on an innocuous looking website that has not been updated since it was first published in 2001 (Manifesto for Agile Software Development, 2001). Originally created by 17 software developers who met in a Utah ski resort to discuss problems they were having with software development, through deliberation the developers figured out what aspects of project management strategies were leading to success. The website was the document they created to share their findings. While initially intended for software development, these principles have now been repeatedly been applied to nearly every industry, sector, and country in the world.

The four Agile values, as stated in the manifesto, are:

- *Individuals and interactions* over process and tools;
- *Working software* over comprehensive documentation;
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan.

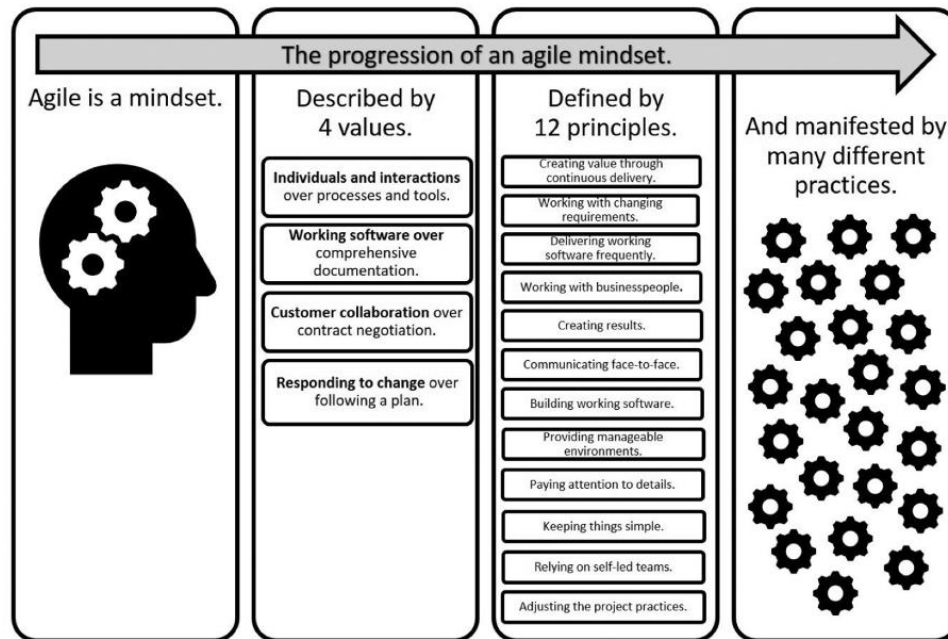


Figure 2. The Agile mindset, values, and principles. Source: Adapted from (Ragas & Ragas, 2021)

The 12 principles of Agile simplified by expert on Agile (Ragas & Ragas, 2021):

1) create value through continuous delivery; 2) work with changing requirements; 3) deliver working software frequently; 4) work collaboratively with business people; 5) create results; 6) when possible, communicate face-to-face; 7) build working software; 8) provide manageable environments for teams; 9) pay attention to details, including technical excellence and good design; 10) keep things simple; 11) rely on self-led and self-organizing teams; and 12) the team adjusts the project practices as needed.

Modifying a business practice in pursuit of coming closer to a value or principle is considered an Agile transition. Many companies will adopt all principles, some companies will only adopt one. Each principal and value is core to an Agile mindset. The principles were originally designed for software teams, but can be scaled and modified to suit organizations of all

sizes, most business practices, and all industries. The values and principals are simple to read and understand, but often challenging and time consuming to correctly implement. For an organization to become truly Agile, they must change the way all of their members think, a seismic shift in mindset that is extremely difficult to execute.

The most difficult challenge for many firms thinking of switching to an Agile mindset is delegating down responsibility and the creation of a more egalitarian workplace. The best Agile teams are self-led and staffed with project managers and/or ScrumMasters taking on an alternate role to their traditional place in the hierarchy. The role of the manger switches from of being a direct superior to removing impediments to the team's success and creating a positive work environment (Rigby et al., 2018). An excellent Agile team is self-regulating, adaptable, and motivated. The team provides from within what would have traditionally been enforced from the top.

Chapter 2

Agile Implementations

Almost all organizations that adopt an “Agile” mindset, do not use Agile as it is. Instead, they implement an Agile Framework. Each framework translates Agile values and principles into actionable items and best practices that ease implementation and adoption. Several project management frameworks have been created including: Scrum, Kanban, eXtreme Programming (XP), the Scaled Agile Framework (SAFe), DevOps, Crystal Methods, Scrumban, and Test Driven Development (TDD). Each has benefits and costs and are tailored to solve or improve specific aspects with Agile implementation. For example, The Scaled Agile Framework (SAFe)

attempts to make it easier to scale Agile on an enterprise level by combining Agile and Lean. In SAFe, an implementer first finds the value streams for the entire company, then organizes groups of Agile teams around those value streams to optimize delivery and efficiency (Linders, 2015). This chapter aims to briefly define two of the most popular Agile frameworks, Scrum & Kanban.

Scrum

Scrum is an Agile framework created by Jeff Sutherland, Ken Schwaber, and Mike Beedle (Carvalho & Mello, 2011). Its namesake came from a classic study by Takeuchi and Nonaka (1986) in the Harvard Business Review, in which developers are compared to rugby players. The Scrum being the quick meeting players have when they are about to start a move, and all pushing in the same direction with small multitask teams. Scrum establishes a set of rules and practices for project success.

These practices include (Carvalho & Mello, 2011):

Product backlog; Daily Scrum; Sprint; Sprint Planning Meeting; Sprint Backlog; Sprint Review meeting.

The *Product Backlog* is the prioritized tasks to be completed or fixed. They can be everything from updates requested by the customer, unresolved support tickets, to new features the team wants to implement.

The *Daily Scrum* is a quick meeting that all team members use to touch base at the beginning of the day to keep each other updated and ensure a consistent but manageable workload. It is also referred to as a *Daily Stand Up*.

A *Sprint* is the primary practice of Scrum and one of the most widely adopted Agile tools. It represents the smallest organizational unit of planning in Scrum. It is commonly set at two weeks, but occasionally varies from one to four weeks. These are the units of time in which the team plans tasks. If a task in the backlog is too long to be completed in a single Sprint, then it needs to be broken down into smaller manageable parts.

A *Sprint Planning Meeting* is held at the beginning of each Sprint. There the Scrum team will decide which tasks from the backlog, usually the highest priority, will be accomplished during this Sprint. Those items are moved to the *Sprint Backlog*, or the prioritized tasks to be completed this sprint. *Sprint Review Meeting* is the debrief meeting after each Sprint where the Scrum team can discuss what went right and wrong.

The *ScrumMaster* leads meetings and removes non-technical obstacles from the team so they can stay focused. The *Product Owner* is the one member of the team designated to represent the external or internal client. They provide insight on how tasks should be prioritized within the Product Backlog.

As a longitudinal example, a given Scrum team will have the four quarters of the year. Each quarter has six, two week, sprints. Each of those six sprints has roughly 10 working days. It is the responsibility of the team and ScrumMaster to make sure that no tasks are added to the Product Backlog or in any given sprint that cannot be accomplished by the team in 10 working days (Project Manager/Scrum Master, 2022). Figure 3 shows a rough overview of a Scrum cycle.

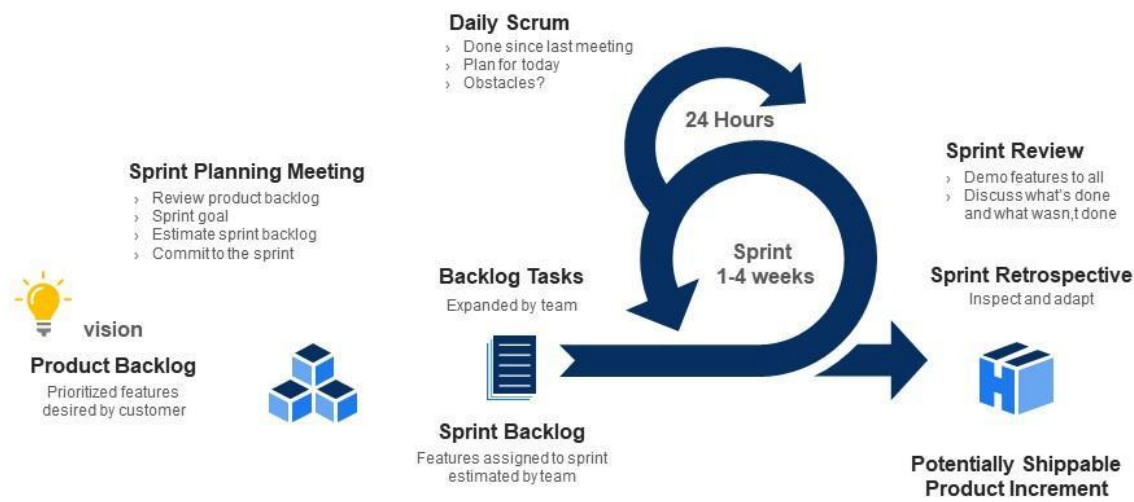


Figure 3. General Idea of the Scrum Process. Source: (Carvalho & Mello, 2011)

Scrum is designed for smaller “Pizza sized” teams between 5-8 people. The name given by the rough maximum amount of people that can comfortably split one large pizza. When teams get larger than 8 people, they can tend to have lower efficiency as the communication process can break down between larger groups. A larger team is also less nimble as more people are required for consensus and input at every step in the Scrum cycle (Project Manager/Scrum Master, 2022).

Kanban

While often considered an Agile Framework, Kanban started before the Agile as a scheduling system to support Lean manufacturing. Kanban translates to “signboard” and refers to a literal signboard in a factory that would be used to track parts as they moved through Toyota production. Developed by Taiichi Ohno, an industrial engineer at Toyota, It has since

transitioned beyond the auto industry and is often regarded as a critical Agile tool, often being used in conjunction with other Agile frameworks (Huang & Kusiak, 1996). As an example, Scrum & Kanban when used together can be referred to as Scrumban. An animated Kanban board is depicted in Figure 4.

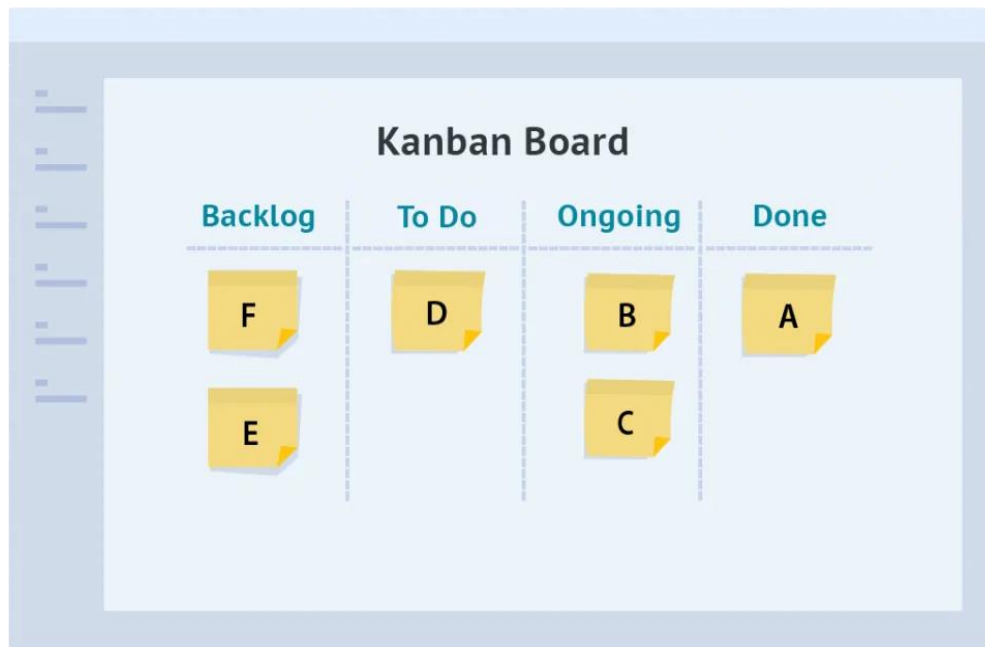


Figure 4. A Representation of a Kanban Board. Source: (Digite, 2021)

In modern Agile implementation, a Kanban board is used as visualize a team's sprints. It allows a team to easily track who is working on what, and how much is being worked on at any one time. These boards were and still are physical, but many online tools have been produced like JIRA, Trello, and Azure DevOps to allow a virtualization of the same board.

Chapter 3

Communications Literature

Despite the proliferation of Agile with regards to organizational management in other fields, Agile and Agility are only barely becoming noticed within the field of communication research. When using the individual search terms Agile, Scrum, and Kanban in the database *Communication & Mass Media Complete* only nine hits were found pertaining explicitly to Agile frameworks. The most substantial contribution provides a thorough overview of Agile for Strategic Communicators by Matthew Ragasa and Traci Ragas (2021). Since the publication of the Agile Manifesto in 2001, communications research was stagnant for many years before the first article was published in 2011. Since then, interest in the literature has picked up. Three articles were published in 2015, one in 2018, and four in 2021. While still currently limited, as Agility becomes ever more relevant to organizations in every discipline, it is likely that communications research on Agile will continue to expand.

Of the nine papers analyzed, two provided broad overviews on Agile product development itself (Carvalho & Mello, 2011; Ragas & Ragas, 2021). One emphasized the importance of Agile to the future of strategic communication (Dühning & Zerfass, 2021). Two presented ways that Agile and its methodologies could be taught in a classroom (Opt & Sims, 2015; Pope-Ruark, 2015). Two describe how Agile is necessary and relevant to public relations and corporate newsrooms (Seiffert-Brockmann et al., 2021; van Ruler, 2015). Of the nine, only two were quantitative studies that used data to make conclusions about a change in language. Friess explored how project manager language is affected by Scrum (2018) and Friess & Lam looked at how implementing Scrum might affect measured levels of empathy within student groups (2021).

Additional literature is needed to explore Agile properly, and resolutely within the context of communications research. Agile is currently in the process of reshaping the world of project management and organizational communication. Further papers, principally those with a quantitative approach are needed.

Chapter 4

Analysis

Agile is on the way to becoming the dominant project management force for the global stage. Aspects of Agile are under consideration and implementation in nearly every country and within the Fortune 500. This section briefly discusses aspects of Agile that make it unique compared to other modes of workplace organization.

Mix & Match

Agile is anything but consistent in its practice. While the values and principles are immutable, how an individual or organization executes on those is down to personalization on behalf of the entity. The endless combination of different frameworks allows entities to be Agile in their execution of Agile, as none are required for correct implementation of the mindset. But it just as easily goes the other way, “Just because you feel like you can think flexible or give your employees flexibility. And, you know, just because you put them in a couple pods together, so they could collaborate that didn't suddenly make them Agile” (Project Manager/Scrum Master, 2022). Agile is not something that can be implemented once. It is a constant process to uphold the values and principles in the pursuit of a modern way to organize teams.

Purpose of Waterfall

Even in a fully Agile organization, Waterfall not an obsolete means of project management. Waterfall is perfect for projects that are both simple and repeatable. Agile was introduced to help solve complex issues, but not all problems and projects are complex. Where simplicity and repeatably are present, Waterfall should remain in practice, as it will be the easiest and most robust way to efficiently provide solutions to those problems.

Flexibility & Robust Teams

Agile lends itself to incredible levels of flexibility not available in traditional Waterfall models. Teams are self-organizing and often multifaceted; this allows them to better tackle challenges and unforeseen roadblocks in the value pipeline. This can be as simple as someone getting sick. In a Waterfall model, if someone gets sick and no one has been trained on their position for redundancy, the entire system halts as a key cog is now missing. Agile's strength in prioritizing adaptability and flexibility at the smallest level of an organizational, lends Agile teams the versatility to adapt.

Democratization & Empowerment

A constant theme revealed through interviews and primary sources is that of the importance of democratization in the workplace as a prerequisite to Agile. Both interview participants had significant direct lines of communication with almost all levels of C-suite in every role they had experienced. The developer mentioned, that when they first got to their

FAANG company, they sent an instant message announcing themselves to the CEO of the company and they received a direct reply. It is important to note that a company culture where a new non-management employees have direct access to a CEO is far outside of traditional management practice. Democratization represents a seismic shift and how employees are treated and recognized within organization. Empowering the worker, and the worker's team, to prioritize and choose their own backlog is a complete revolution of management methodology.

Chapter 5

Methodology

For the literature analysis, Agile, Scrum, and Kanban were used as independent search terms in the database *Communications and Mass Media Complete*. All relevant articles were subsequently analyzed for content and compared with best practices, external sources, and used to inform the overall conclusions of the paper. For the case studies, two participants with significant experience in Agile were recruited for the case study aspect of this paper. Each participant fell into a distinct group that was designed to represent different aspects of an Agile based organization.

Participant one was chosen to represent the developer or worker perspective of Agile, as in what it is like to work under and within an Agile organization but not lead it. They are in their early thirties.

Participant one has a long history of working under organizations that were semi-structured in their Agile methodologies, and work-spaces that are highly structured based on Agile practices. Participant one has three years of experience as a software engineer building

contracts for the Navy in a semi-Agile environment. Participant one has two years' experience working for a highly Agile machine learning based software startup, that was then acquired by a FAANG company where they are currently based. In these roles participant one has served as both a software engineer and a senior software engineer. In one of their roles, they undertook the title of ScrumMaster and were responsible for running daily and weekly meetings for their team. Participant one has a four-year degree in computer science and is a reputable authority on machine learning.

Participant two was chosen to represent management and the project manager role in an Agile organization. Their experience represents a microcosm of the mindset behind how project managers, and other higher-level stakeholders, might approach Agile and its workplace philosophies. Participant two is in their early 40s.

Participant two has served in three project management roles at different companies. In each one they oversaw one to three teams and were responsible for deliverables and organization. Participant two became more involved and directly affected by Agile in each sequential role. The first company was Waterfall based, the second company underwent an Agile transition during their tenure, and the third company was already fully implementing Agile practices (Scrum, Kanban, XP) prior to their employment. Participant two also holds a four-year degree in industrial engineering and four Agile certifications. They are certified ScrumMaster, a certified advanced ScrumMaster, and a certified Scrum professional. Each of these certifications are issued by the *Scrum Alliance*. They also held a scaled Agile framework certification from the *Scaled Agile Academy*. Each of these certifications requires several days of in-person training from the respective licensing organization.

Chapter 6

Interviews

FAANG Developer: Initially, participant one described Agile development to be “Doing what's needed as fast as possible and as flexibly as possible.” When prompted to elaborate, additional robustness was not directly forthcoming, as to them, every work environment they had been in, had always been structured according to various Agile principles. Participant one, being younger than participant two, entered the software workforce by the time Agile principles had seen widescale adoption. “Personally, pretty much every job I've had has been an Agile environment.” It was hard for participant one to separate out what were standard workplace practices, and practices they would define as Agile. Under further prompting, participant one begin to label and contextualize the impacts that Agile had on the work environments that they participated in. In the participant’s words summarizing the omnipresence of Agile, “it (Agile) has become so ubiquitous that it is it is no longer special. If you find an Agile environment, it's like ‘oh, that's not really special.’ They're just doing it the way they should be.” Participant one also alluded to the lack of consensus on Agile, “I think if you sat a room of 100 engineers down and ask(ed) them what Agile meant, you'd probably get a different answer from all 100 of them.” Participant one recognized that the lack of consensus is not necessarily indicative of a negative aspect of Agile, as teams perform differently, and unique Agile practices can be implemented in a non-uniform manner.

With regards to education, participant one has limited experience with formalized training, despite their previous role as a ScrumMaster. The most formalized training they

received with respect to Agile, was for tools like JIRA. They were taught how to use it, and how its proper use would bring their company closer to Agile principles. For their current job at FAANG, when they were on-boarded they had to take several standardized companywide trainings (Bootcamp) that mentioned Agile, and Agile methodologies, but they were not specific to any one implementation, “It's more of like, this is how we do things, we want you to be able to do them too. So you can fit in here and do your job effectively.” Participant one noted that apart from extremely linear tasks, Agile provides significant structural advantages.

When prompted on the topic of workplace equality, participant one noted there have been significant improvements over the years, especially when personally evaluated against their parent’s anecdotal work stories. They did correlate this with Agile, but were not robust on the perceived causality, “We've finally reached the point where, again, there's more of this equality between managers and the people working under them. I don't know if that's as much a result of people finding that Agile works better as it's just like, a wider change, in workplace culture over the years.” Participant one noted that they have always had nearly unfettered access to every aspect of the chain of command within the companies they have worked at. “At my last role, I directly interacted with the CEO and CTO, and CEO as necessary. And I interacted directly with stakeholders as necessary. In my current job I interacted directly with the CTO and CEO, and, like head of HR directly.” In participant one’s experience, ideas were communicated more effectively, and problems addressed quicker. As opposed to if a more traditional chain of command had been present. This ease of communication, while not concrete on causality with Agile, was required in participant one’s perspective, for proper implementation of Agile, “you that sense of equality and openness.”

FAANG Project Manager: Participant two described Agile as a mindset and was firm on the definition throughout the interview. To participant two, “I view it (Agile) as more of a mindset than anything, it’s definitely not a framework. It’s definitely not a toolset. So I view it much more into the mindset area.” Participant two’s understanding of Agile began with the Agile Manifesto and its core beliefs and principles. From the perspective of participant two, regardless of extraneous factors in a business or team, as long as each individual member could read the Agile Manifesto and demonstrate their self-adherence to those beliefs, they were working in an Agile manner. This process within a company can be aided by frameworks.

With respect to Scrum, participant two views “Scrum as a framework that specifically builds on that mindset and allows you to have a repeatable process day-in and day-out to actually be more Agile.” In the participant’s current company, they manage two Scrum teams each containing approximately eight people, as that was determined to be the optimal workload. As a ScrumMaster their job is to monitor progress, mediate meetings, and assist with whatever is necessary to make the teams as productive as possible. Participant two was adamant in viewing their role as that of a facilitator, not a leader, “I have the autonomy to figure out the methods that are going to work for what I want to propose back to the team. But the key for me is: I want it to be much more facilitation than dictatorship. I never want to tell the team what they have to do. I’m there to facilitate conversation,” as they also viewed equality as central to the core function of Agile.

In regards to education, participant two holds several credentials from multiple certifying organizations, each of them with an Agile like approach for implementation: small group collaborative workshops with instructor guidance. They were shown primary source resources, but participant two is highly skeptical of any empirical research in the project management space

and is relying primary on his own personal experience, “The reason I haven't shifted (back to Waterfall) when I went to Agile 10 years ago is because it works a heck of a lot better for us.”

Participant two has also attended several Agile conferences including yearly attendance at *Agile and Beyond*. In this conference, they are exposed to the wide-ranging applications of Agile across the business space. In their view, “the applications of Agile, when abstracted are all very, very similar. But their circumstances are very much different.” Participant two acknowledged that it was not universal. Questioned for specifics, they described how Agile is best for “simple repeatable projects ... (because) it's easier for a client to understand something when they feel like they have a fixed a fixed project where they feel like they know how much time it's going to take, (and) they know how much money it's going to cost ... Where it starts to fall apart is when you have complexities in there.” Agile is beneficial for solving challenges that arise via complexity, but it does complicate the business process in the areas of contract negotiation, and client - firm relationships. They reiterated that these are challenges with implementing Agile in any business setting, and it will not always work for a given project, but the core of Agile is being flexible. Thus, you must be Agile in your implementation of Agile.

Participant two repeatedly emphasized the importance, of self-governance and regulation within Agile teams. In their view, for a team to be Agile, they must be setting and prioritizing their own agenda and workflow. Granting and ensuring this autonomy is preserved, is paramount to their role as a ScrumMaster, “Autonomy is as powerful thing to an engineer, as it is to a project manager. It's you job to manage your own destiny, having that recognized makes you feel valued, and it's exciting.” An added benefit to the self-governance is “a much more sustainable pace of work” for each of the teams because they only assign as much work to themselves as they can do in a given sprint, and if they do decide to push, it's not management's decision, it's theirs. This self-governing agency and flexibility are fundamental to participant two's framing of Agile. It is these principles, not the frameworks that matter, “Scrum is not in and of itself Agile. If you just do Scrum, and don't understand the essence of Agile, you'd still not be Agile at all.”

Chapter 7

Discussion

The interviews in the study were designed to give a broad overview of the participants understanding and experience with Agile. The interviews started with the same question sheet but had diverging paths as the interviewer probe specific areas of expertise. While only two interviews are represented in the study, they are reproduced in the appendices in full. Communications literature on Agile is already limited. Both interviews are reproduced to fortify access to accessibility to primary source opinions and analysis of Agile. Every effort was made

to reproduce the exact words and intentions of the speaker, but for readability, clarity, and brevity certain sentences may have been altered, and should not be taken as truth. Due to the anonymous nature of the interviews, their use to further research may be limited.

The communications literature on Agile is extremely limited and should be significantly expanded before any robust conclusions can be drawn about Agile from communication scholars.

Chapter 8

Conclusion

In the environment of the 21st-century business world, embracing adaptability has become essential to the survivability of a company. Agile and its frameworks continue to play an ever-increasing role in the modernization of project management. Building on past project management advances like Lean and TPS, the software developers responsible for the Agile Manifesto were able to innovate and isolate the core of what is now known as “Agile.” Starting within the software development space, but now spreading to other industries, the Agile mindset continues to redistribute power within organizational hierarchies, create resiliency and redundancy within teams, and improve both productivity and quality of life for employees. While implementation is never the same between companies, if done correctly, the core benefits may be still be derived. Frameworks can be used to align organizational goals with Agile principles, but without a mindset change, an organization will never be Agile. For companies to survive the turbulence created by the paradigms of modern business, Agile is not optional.

Appendix A

INTERVIEW QUESTIONS

1

- Describe your role in current & past interactions with Agile.
- In your own words, define “Agile.” Mindset, Philosophy, Framework, Toolset, etc.
- What forms of Agile do you have experience with? (Scrum, Kanban, XP)
 - o Describe your relationship with the practice of Sprints? (Participant 1)
- What is your relationship with Lean / 6 - Sigma?
- Have you ever worked in a Waterfall environment?
- How was/is working in an Agile environment different from a Waterfall environment?

2

- What did you think of Agile while you were a worker vs now in management?
- What kind of Agile training have you received? How has it changed the way you approach the topic and management in general?
- Speak to the resources available within your training?
- Were you ever shown peer-reviewed/scientific resources?
- Was there ever a lack of consensus on what Agile meant or how it would be introduced?
 - o Is there any conflict as a result? (Participant 1)

3

- Have you ever led a team in a Waterfall environment?
- Would you be able to speak to the effectiveness of your team in both Waterfall and Agile?
 - o Does it affect the sense of time pressure for completion?
- Are there products or services that make more sense to be managed in either a Waterfall or an Agile way? Does scale play a role?
 - o What is Agile management like compared to working under an Agile team? (Participant 1)
- How do teams negotiate fixed contracts for services or products while working in an Agile way?

- Does Agile make more sense for subscription or app products? (Participant 1)
- Agile claims to significantly improve customer satisfaction and productivity. Is this true in your experience?
- Do customers want to participate in the Agile process? What if they don't? (Participant 1)
- In the interest of attaining a gold standard hybrid model between Waterfall, Scrum, Kanban, etc. What specific methods or behaviors are the most significant in delivering value?

4

- At its core, Agile is about delivering “value.” But, “value” can mean different things to different people on the same project. Do all involved stakeholders have the same definition of “value” with reference to Agile and the workplace services?
- What do you do if the customer doesn't participate/collaborate? How often does this happen?
- Has increased customer participation and rapid prototyping ever resulted in reduced productivity? Rushed designs?
- What was the transition to Agile like in the programming and project management space?
 - How Agile affect a corporation's hierarchy? (Participant 2)
- When you see Agile Lead or ScrumMaster on a job description, information does this convey about the role or company?
- Are there any other significant benefits/drawbacks to the Agile model that we haven't already discussed otherwise?
- At the end of the day, do you feel there is anything special about Agile based project management?

Appendix B

INTERVIEW TRANSCRIPT – FAANG DEVELOPER

Jacob Iwinski 0:02

Hello, my name is Jacob Iwinski And this is Participant 1. Would you describe your current role and current and past interactions with Agile?

FAANG Developer 0:15

Sure, my current role is I am a software and machine learning engineer for a FAANG company. I work under the business and messaging group as part of customer. And my current and past interactions with Agile all involve software development and machine learning development. And using the Agile methodology and doing so.

Jacob Iwinski 0:39

Could you please just list maybe some roles that you've had that directly involve Agile?

FAANG Developer 0:45

Sure. The primary one that I've had has been that of, I guess an employee is like partaking in Agile. I guess I don't know what the proper term would be for it, I guess, as a user almost. I have in the past, acted as a ScrumMaster. I don't know what the exact term for it is, but like managing something akin to a Kanban board. But the vast majority of it has just been like someone following the Agile methodology for development.

Jacob Iwinski 1:38

Thank you. Question two, in your own words define Agile.

FAANG Developer 1:47

In my own words, I guess it would just be having a very flexible approach to development as opposed to something that is extremely linear, like the Waterfall model that came before. And a lot of older and larger companies still follow with development. I know there's a whole bunch of like, methods of being Agile, like Scrum and things, but I just consider I consider Agile development to just be doing what's needed as fast as possible and as flexibly as possible.

Jacob Iwinski 2:32

That falls more under, more a mindset or philosophy than it does framework or set of tools.

FAANG Developer 2:40

Yeah, because there's, because that's one thing I have encountered personally, pretty much every job I've had has been an Agile environment. When I did work for the Navy, we had to prepare our deliverables in a more Waterfall approach. But we internally followed Agile methodologies. But there wasn't really there wasn't really like specific tools that we used. We didn't use a perfect Kanban board. And we didn't follow perfect Scrum procedure. We just adapted those philosophies or our processes, to what works best for us.

Jacob Iwinski 3:38

So you grabbed say, certain parts of it, like maybe having a daily standup? Or maybe having a debrief, maybe having a sort of Kanban board?

FAANG Developer 3:48

Yes. One thing that I've used very frequently is a tool called JIRA. Which is Kanban. Ask, I don't know if there's another. I don't know if there's another Agile tool that's closer to what JIRA does than a Kanban board. But that's like, the only specific Agile tool I can say that I've used.

Everything else has been note taking during Scrum meetings and daily stand ups. For quite a while I lead the daily stand ups. Currently, I'll do stand-up meetings, but not daily because it's not conducive to the way my team currently works. Defining and following Sprint's is another thing that we've all of all of my jobs have followed to some degree. What constitutes a sprint has been vastly different between the roles that I've had.

Jacob Iwinski 5:37

Could you talk a little bit more about that sprint, like maybe give two examples of when a sprint was different and how it was followed at those companies?

FAANG Developer 5:44

I'm to think of what I am allowed to say, because it was for the Navy. In preparing deliverables for the Navy, our sprint was effectively the period of time we had between deliverables because we had a single stakeholder. So we defined the sprint as the entire amount of time between, one deliverable and the next. In my other roles, a sprint as has been a set period of time, typically two weeks and following the sprint is not so much. Having a set deliverable ready. It's you pull tasks into the sprint, you assign them points, and you discuss with your team like what do we think we can go through during the sprint, and following the sprint is more or less following that plan that you discussed. Like if you pull X number of tasks into the sprint, and you only finish half of them, you are not following the sprint very well and you need to adjust what you have pulled into it. Whereas before when we were preparing deliverables, the sprint was the amount of time we need this done by this time. You better have it done.

Jacob Iwinski 8:10

Question three, and we kind of talked about this a little bit, but what forms of Agile do you have experience with as far as Scrum, Kanban, Extreme Programming, etc. If they were broadly defined at all?

FAANG Developer 8:28

I would say Scrum and Kanban obviously

I'm just looking over a list of things to make sure that I'm not forgetting anything I've worked with in the past. Feature Driven Development would be one that I've worked with in the past I think.

I would never call it this, but I guess it's called Extreme Programming. I would never call it that ever. I've forgotten a lot of these terms. Lean software development. Rapid application development. Those are probably the main ones that have set names.

Jacob Iwinski 9:58

Could you talk about your relationship with Lean software development and or Six Sigma.

FAANG Developer 10:18

It's not like we sat down and decided, like we're going to follow like this exact rule book or anything, it's more of just following a lean methodology was just necessitated by the size of the team, like one of my roles, I was the only person working on the project. So I could not be bogged down by doing anything other than what was absolutely necessary. So in cases like those you are more or less forced into following a lean design and software development methodology, because if you try to do anything else, you're going to commit to far more than you can deliver.

Jacob Iwinski 12:29

Question four, have you ever worked in a Waterfall environment?

FAANG Developer 12:35

So sort of, like I referenced before, are, but when I when I was working for government contractor doing work for the Navy the nature of a lot of government contracts leads to at least our interactions with them, following a, a more Waterfall approach to development, where there are like, what's the word there's, like, dedicated, like steps to the lifecycle of what you're building and delivering to them. But the company I worked for, did not follow that approach outside of how we delivered things to the stakeholder.

Jacob Iwinski 13:27

Was working in an Agile environment different from a Waterfall environment?

FAANG Developer 13:33

I think the main thing I noticed, between how we did things and how we would end up delivering things to the to the Navy was kind of continuous development and testing, but not like Agile. In Agile for instance, if you know the expected behavior of something, you can write tests for it. And then if those tests fail, then it's not an issue with a test. It's an issue with development, you need to go back and adjust what you have developed to fit the tests that you've written. Whereas if you follow a very strict Waterfall approach to things, the tests go at the end. And I mean, technically you still must go back and change things if the tests fail.

I think Waterfall is more of a byproduct of back when the things that were being developed are physical and not necessarily like software.

Jacob Iwinski 16:01

One of the aspects of Agile is writing tests, as opposed to writing requirements, essentially making the tests, though requirements. So that makes a lot of sense.

FAANG Developer 16:14

Yes, and that's another one of those things where I don't think I've ever actually seen someone follow that to like a tee of writing the tasks instead of writing requirements, mostly because the entire world are not developers. Unless you are developing a very specific tool. As a developer

for other developers, like if you're more effective, if you're building software for people who are also building software, then I could see where that would be something that would be followed very closely. But a lot of the time, it's very non-technical people, giving technical people a set of requirements. And like, they're not going to be like your and in those cases, you're usually sometimes you might, you might want to start by writing the test. But usually, there's a whole lot of work that would need to be done way before ever writing tests.

Jacob Iwinski 17:18

Thank you. Question six. What do you think of Agile as a worker versus in more of a managing position? And how did that change your perspective?

FAANG Developer 17:30

Yes, so the only time I've worked with Agile in a management position, it has been primarily as a ScrumMaster. And as effectively the technical lead. When I was working for the government contractor delivering to the Navy, and there was kind of a kind of a hassle having to like lead daily stand ups and things. But having never, you know, managed outside of a Agile environment, I'm not sure like, I don't have anything to really compare it to. As a worker, I don't know that I would ever want to work somewhere that did not at least follow the philosophy of an Agile environment. Again, I don't know many places that are really strict about following exact practices and frameworks and, and things from Agile. But I could not see myself going to work somewhere that that like is adamantly against that kind of workflow. Like, we're like an Agile development, or an Agile philosophy, as a worker, I think is kind of key to keeping your sanity and software development. Like knowing that there's that flexibility to go between different knowing that well, just from just from the phase of the development lifecycle, knowing that there's that flexibility to go back.

In a lot of cases, especially for software development, a lot of the Agile development practices, just kind of make sense. It's just things where like, if you're following common sense, you wouldn't want to do them that way. So I could not see. Really not being an Agile in an Agile environment as a worker

Jacob Iwinski 20:01

Thank you. Question seven, what kind of Agile training have you received? If any? And how has it changed the way you approach the topic and being a ScrumMaster.

FAANG Developer 20:15

So the only real direct training I've received for Agile has been in two places. One has been for specific tools, like I referenced using JIRA. It wasn't like a really strict training, but it is a very robust tool. And so I did have to learn how to use it. And then in my undergrad, I don't even remember what class it was that had us go through and learn a whole lot of, you know, a whole lot of the Agile terms and methodologies, frameworks, the whole philosophy of being Agile. But that's really, it. I don't know. How many places focus too heavily on that. I think in a lot of cases, what you'll run into are companies that kind of formed their own development culture that's kind of derived from Agile, and you'll get some kind of training on that.

The company I work at FAANG, has people go through something they call a bootcamp, which does not hammer Agile methodologies in your head or anything. But it is about, you know, introducing employees to their culture, which their culture does include, you know, Agile methods and Agile philosophy. It's not specifically Agile training, though, it's more of like, this is how we do things, we want you to be able to do them too. So you can fit in here and do your job effectively.

Jacob Iwinski 22:37

I think that kind of covers Question Eight as well. What resources were available to you during your training? Oh, yeah. Were any of those resources in boot camp or at an old job, peer reviewed or scientific resource? Or presented as such?

FAANG Developer 23:07

When I went through undergrad, absolutely, I think they were all. Like, directly pulled from, from like, white papers and things, the stuff that they taught us. At the FAANG boot camp, I'm not entirely sure. Because again, it wasn't like, they were specifically trying to teach people Agile methodologies, it was more just like, here's our development culture.

Jacob Iwinski 23:51

Was there ever a lack of consensus on what Agile meant? Or how it could be introduced at any of the companies you work?

FAANG Developer 24:02

Yes, constantly. I think if you sat a room of 100 engineers down and ask them what Agile meant, you'd probably get a different answer from all 100 of them. It's one of those things where I am not aware of anywhere that it that really strictly follows, like the specific like, specific practices, frameworks and things. to a tee. It's kind of I think, pretty much everywhere has just kind of taken it taken what they've wanted from Agile, adapted it to fit, how they work, and made it their own. And so I think pretty much anywhere you go and anyone you ask, other than, like maybe a professor at a university will probably give you a similar answer, but probably not the exact same one.

Jacob Iwinski 25:03

In your experience, where there was there ever any conflict due to disagreements on what Agile meant?

FAANG Developer 25:14

Specifically, on what Agile meant? No, there have been conflicts on what would be like the best use of certain like the best use of things like Scrum. There have been conflicts about what wave what way of doing something like Scrum or Kanban board, stand up meetings, etc. Like, different disagreements on like, what way of doing that would be the most useful, not necessarily the most Agile, but like, what interpretation of it or what usage of it would be the most conducive to getting our jobs done. But the focus is the focus is everywhere I've been at pretty much everyone I've talked to the focus has always been more on what way of doing these things or following these things to finding these things causes us to generate the most value.

Jacob Iwinski 26:19

So there would be discussions regarding what aspects or practices of Agile Scrum or Kanban this particular company should be adopting for this team at this time. And there would be discussion on that.

FAANG Developer 26:34

Yeah. There would be discussion on that. Usually, it's less of focus on the on the like, usually, the focus is more on specific tools. Because some things like some things like like a stand up, you're specifically talking about a a practice and Agile practice. But usually, it's been like one of these tools that are available, for instance, JIRA, again. Like what of these should we use? How should we use it? I can't remember what what was used before what we use before JIRA. But the vote Yeah, the focus has usually been on like products more than specific, like Agile terminology and things.

Jacob Iwinski 27:28

Have you ever led a team in a Waterfall environment? Question 12.

FAANG Developer 27:41

Kind of again, going back to delivering things to the Navy, who was our sole stakeholder for that project? I was effectively the technical lead for that project. Though. Internally, we were not necessarily following a Waterfall environment. It was just our interactions, the Navy more or less followed a Waterfall methodology.

Jacob Iwinski 28:14

Question 13 Would you be able to speak to the effectiveness of your team in both Waterfall? And in Agile? In essence, does aspects of Agile or Waterfall impact how you approach and do things? Or does time pressure feel different based on having or not having sprints for example?

FAANG Developer 28:39

Yeah, I would say following a Waterfall approach. Especially in software just kind of feels like it almost feels wrong. It's very rigid. It kind of like in our interactions with the the Navy and kind of following a more Waterfall approach. That was the one time where it kind of felt like you were going back to going back to the days of like all the software engineers show up in in like, a buttoned up white shirt with a tie and dress pants and and like nice dress shoes and crap like it almost feels like you're going backwards in time following that kind of a methodology. Whereas at trial just kind of it just feels like when you're following more a more Agile approach using using you know, Agile tools and following a more Agile philosophy just kind of feels like you're doing things the way that makes sense. I think again, I think Waterfall in the Waterfall methodology comes from a time where deliver Have goals and products were more physical than the kind of abstract products that a software engineer and a software company provide.

And it's not. And it's not as detrimental to go backwards. Because I mean, if you think about where the Waterfall methodology comes from having to go back and change something, when you're at the very end, could mean like, millions and millions of dollars wasted easily. If it's if it's like, you realize, oh, there's this massive problem in a, in a manufacturing pipeline, versus with software development. If you get to the end, and you realize there's something wrong, there's a

decent chance somebody just screwed up some logic somewhere, and you can take the like day or two to fix it, and you're good. And you're not out. Like you don't have to necessarily scrap everything that's been built since that that issue was introduced. So I think they're I think they're there too. They each have their place.

They are geared towards two very different things.

Jacob Iwinski 31:28

Excellent. Question 14. Are their products or services that make more sense to be managed in an Agile versus Waterfall way? Or vice versa? And does scale of the company play a role?

FAANG Developer 31:43

Absolutely, well, like I said, if you're working on something where a screw up early on means that everything after that point, would also need to be scrapped to correct it, then Waterfall would really be the only way of managing something like that. And software, that's very rare, it'd be very, very rare that that an issue introduced early on cannot be simply fixed later, the only thing I could think of would be the only thing I can really think of would be a few ended up finding out that like, the entire language would program something and couldn't be used at the very end. And you'd have to translate it all to an entirely new language or framework. But otherwise, I would think pretty much anything software related should be developed in an Agile way. As far as scale goes, I think the fact that companies like Meta and Amazon and watch I'm not entirely I think Amazon mostly follows that Agile philosophy. Google, Netflix Bing is like those super massive companies that deliver products, well, software products to billions of people a day, are able to follow Agile, I don't think scale necessarily plays a role in what's better, or what's necessary, because at the same time, you also have massive manufacturing companies that deliver tons of products to people each day that follow a more Waterfall methodology. I think it's more or less what you're developing that determines what methodology you would follow and don't think the scale really matters.

Jacob Iwinski 33:40

Thank you. Question. Well, this is another subset of question 14, A, so many companies have announced or recently or previously that they are transitioning to a fully Agile environment, not just individual teams, but the entire company, or you know, even management at that company. So could you speak to Agile on like a team by team basis versus Agile on a management or C suite basis and how that might differ in their implementation and understanding?

FAANG Developer 34:18

I have not spent a lot of time managing people. So as far as what the alternative to an Agile environment as far as management goes, I wouldn't have a whole lot of insight to that. I could see, especially in software companies. Communication is key. If there was a company that previously had had a very set structure. Something like Tesla comes to mind where Elon Musk's word is law. And what he hands down goes straight down the chain and usually does not go back up. That would not be a conducive management environment for something like software development, where if there's an issue, someone just saying, like, I don't care I want it done this way, is going to cause a whole lot of problems. So having an environment where there's communication, both directions up and down the chain is very key to a successful business. It's

delivering software services. But outside of something like that, I would I'd have to see specific examples to give more insight.

Jacob Iwinski 35:43

Thank you. Question 15. How to teams negotiate or price like fixed contracts for services or products, while also working in an Agile way?

FAANG Developer 35:56

Interesting, I think it depends on that. That's another thing that depends heavily on the stakeholder. So if you're working on something like government contracts, it doesn't matter how you work. The way you you negotiate and price contracts is how the government wants it done. And that's the beginning and end of it. For companies that do consulting work, I do believe it's usually based mostly on the number of hours, I don't think it matters whether whether they're following Waterfall, or Agile or any kind of methodology, the main way they price it is like hours worked. And then when you get to more B to C things, I think it's that the term contract, it's a little bit fuzzier. You're more mostly pricing? Because it would still be a, I guess, in that case, it would be a contract would be like a subscription at that point. It's like, what, what can you do to cover the costs of developing the thing and still generate profit? In a lot of cases, and I mean, more and more, the cost is zero, because the users are the product themselves. That's an increasing philosophy. But again, it's I don't think it matters as much What methodology you're following Agile or otherwise, I think it usually will always come down to how do we how do we cover our costs? How do we pay our people? How do we still have some money after the fact?

Jacob Iwinski 37:57

So Agile makes sense in my mind, from the perspective of say, the user is the product or subscription, right, where I pay \$15 a year for this journaling app. And the team slowly works through the backlog of complaints and adds features, as their time becomes available to just keep customer satisfaction. What about Agile? Or at least Could you talk about perhaps back with the government experience, Agile for you know, a fixed contract where you need to explain exactly what you're delivering upfront, and deliver exactly what you promised?

FAANG Developer 38:49

Yeah, so the biggest thing is constantly breaking down larger tasks and the smaller ones, because that's the main thing because like, if you have if you have a contract to deal to deliver, to think of a good example, so you have a contract to deliver a security product to someone. Depends on you know, what kind of cybersecurity product I guess a good example would be. Let's say you're working on an anti-virus program. You would want to break that down into at the highest level, front end and back-end work. So how do we pull files from the system? How do we scan them? How do we how do we know that they're viruses? And depending on what kind of contract that is, you would be doing things like from the front-end side, sending mockups to your stakeholder, and then getting feedback on them and potentially changing what UI elements look like. From a back-end standpoint, it's a little bit trickier would usually be like, evolved depends on how well defined the contract is. If, if there is a, a very, very set in stone like list of behaviors and expected inputs and outputs from a program, then the only thing you would probably be communicating with them on would be if you needed clarification on one of those like line items in the contract.

But for the most part, anything Agile would be totally separate from your stakeholder who you have the contract with would all be internal.

Jacob Iwinski 41:28

Question 16 Agile claims to significantly improve customer satisfaction productivity, is this true? In your experience?

FAANG Developer 41:37

Absolutely.

Trying to spit something out and call it finished is terrible for user experience. Because you're never going to iron all the bugs out of something ever, doesn't matter how simple the program is, somehow there will be a bug in it. So having a set expectation of this done, is just not a good way to go. Whereas in an Agile environment where or enable an Agile environment where communication goes both ways, you're okay with going back up the lifecycle chain and reiterating on something you can, you know, fix those problems, adapt to feedback, generate something that has more value for the end user. Trying to think of their specific example of the only specific example of the Waterfall method falling through. Without getting into specifics, there was some behavior that was expected that the stakeholder ended up not liking. But by the time they had gotten to it, we were approaching the end of our contract and did not have the time to go back and correct Well, it wasn't even a correction it was they wanted something changed effectively. Like if it was the expected behavior, it was what we had agreed on. But because we were following that very strict linear methodology, they didn't fire they didn't have the chance to give us that feedback until it was too late for us to change it. Whereas if we were following something Agile, they probably would have found out about that much sooner in the process. And we would have been able to change it and make them happier with it. Even though we you know, we were still just following the contract we've been given. So

Jacob Iwinski 44:05

Agile is the pursuit of customer satisfaction, makes the most sense if the customer actually wants to participate and review the prototypes. Do you have experience in customers wanting or not wanting to do that and how does it affect the process?

FAANG Developer 44:26

Pretty much every time the customer wants to see the prototypes. You want to see how it's progressing to know that you're not just firing money into a black hole. When it comes to my experience with government contracts they haven't. It's kind of I'm not entirely sure if it was, it almost seemed like we weren't allowed to send them prototypes outside of when we have like the set deliverable schedule. I don't know if that was specific to that contract. I think with software, you would have to be you would have to be either entirely not technical, or it would have to be such an insignificant thing that if there is an issue with it, it doesn't matter to you, for you to not want to see the prototypes. Usually, it's pretty easy to include your stakeholders in like the development cycle, and pretty much pretty much everything I've ever worked on the stakeholders have wanted to see prototypes, and wanted to see the different stages that that our deliverables were at.

Jacob Iwinski 47:21

In the interest of attaining a gold standard hybrid model, between Waterfall, Scrum, Kanban, etc. What specific methods or behaviors do you find are the most significant in allowing teams to deliver value?

FAANG Developer 47:38

I think is going to be almost entirely dependent on the team? I think the only the only things I could see being universal would be something like something like a Kanban board for tracking the progress of things. And something like periodic stand ups, again, to track people's progress on things. But I mean, what methods and practices and philosophies are going to work for a team will vary wildly by team and wildly by project. There were times where what worked best was me heads down coding 12 to 16 hours a day for up to a month straight without Team interaction because what I was working on something other members of the team couldn't help with. The best thing was just me getting it done as fast as possible.

Jacob Iwinski 49:19

At its core, Agile is about delivering value, but value means different things to different people on the same project. All involve stakeholders might not have the same definition of value with reference to Agile and workplace services.

FAANG Developer 49:36

At the end of the day, pretty much everyone defines value as money. Like any business you run into, doesn't matter if it's for profit or nonprofit. When they are determining the value of something, it will almost always revolve around dollar spent versus dollars generated or saved. Depending on, on what the goal is, but at the same time, at the same time, they also define as value just in the weird kind of abstract sense. People will value really weird things like form over function. There are some places that put the polish and UI value way over the actual functionality of their products. And that definition of value can be very abstract and different between teams.

Jacob Iwinski 51:20

In the Agile methodology, rapid prototyping, and customer participation and said rapid prototyping is critical. Has either of those ever resulted in reduced productivity or rushed designs when you feel like you must have a prototype or something done within a sprint, that should have taken longer?

FAANG Developer 51:44

Yes, rapid prototyping. When you need to get something, when you when you need to have something ready for a stakeholder to review and give you feedback on, and you have a limited time to prepare it. Unless that prototype is something excessively simple is always going to. It's always going to impacts overall productivity just because it's like you need something for them to see. Especially if you're dealing with like software, where what the person sees is just the UI side of it. Rather than having all the actual functionality ready that you could have had, if you had just focused on that. It's like oh, but we need this, this pretty interface for someone to look at otherwise, like nobody cares about. Nobody cares about code. At the end of the day, people care about what they interact with and what they see. And I think that's one of the key places where

rapid prototyping and getting user feedback can kind of bite you in the ass is that you end up focusing on that point of interaction.

Jacob Iwinski 54:12

The transition to Agile in the programming world affected a lot of people now you weren't around for the pre Agile programming or software world. But have you experienced or talked with anyone that still have remnants from that? And how has the shift in Agile, affected? Software and project management in your experience.

FAANG Developer 55:02

So the only person I've talked to directly about, I guess, would be kind of project management from, from, like, pre-Agile methodologies would actually be my dad. He was a manager at IBM years ago.

I think one of the biggest transitions to Agile, is how management is handled. Before you had this hard hierarchy of, managers with all individual contributors below any manager. And then there's just a chain of higher and higher managers. Everyone at the bottom must do whatever the next person up the chain says. And that's it. I think over time with the adoption of Agile, people realize that that system doesn't work.

There's more of a culture of individual contributors are not necessarily lower than managers, and managers are not necessarily, you know, better than individual contributors. And there is a lot more collaboration, whereas for many, many years, the setup was, you go through chains of management, down to individual contributors, and then back up the chain of management. Whereas now, for instance, at my last role, I directly interacted with the CEO and CTO, and CEO as necessary. And I interacted directly with stakeholders as necessary. In my current job I have I have interact, I interacted directly with the CTO and CEO, and, like head of HR and stuff directly, even though there were technically like, two levels of management and between me and them. Because sometimes it's easier to just go directly to the person who, who you know, is making the decisions, and get feedback from them and, and give them input on things. And I think that's probably the biggest change.

My dad told me about one time at IBM the grandson of the guy who started IBM was going to be visiting one of their plants in upstate New York, and that was such a rare occurrence for the highest level of the company to be visiting some lowly plant that they completely remodeled the whole building, in in preparation for his arrival, as if he were God coming to visit them. And nowadays upper levels of management and CEOs will go visit manufacturing plants. Just see how things are done. Where years ago, that just wasn't the case.

Jacob Iwinski 59:58

Do you feel that Agile has this increased kind of equality in the workplace as the dissolution between management and employee has the balance of power has shifted?

FAANG Developer 1:00:26

Depends highly on the company. My understanding is that places like finance and banks are still kind of set in the old ways. If the CEO walks by you don't dare look at him, let alone talk to him.

Where at a company like mine? I have sent direct messages FAANG CEO got a response. And that response wasn't, why are you bothering me? I think depending on what industry you're in, and what company you're at, things have become much more open. I'm actually incredibly impressed with how open FAANG is for as large a company as it is and as valuable company as it is. But I don't think it's a universal change yet. I know. I know, there are a lot of sectors that that even though they might be adopting Agile methodologies, they're still hierarchies with employees and management that have not yet changed. I have a friend who works in consulting, like business consulting. Specifically internal audit consulting, out in Los Angeles. And while they follow somewhat Agile methodologies their like business structure and approach to management is very, like 1950s.

Jacob Iwinski 1:04:46

Do you believe that this equality is a requirement for Agile?

FAANG Developer 1:05:13

Not necessarily. There are plenty of businesses where, where teams effectively develop and deliver products. Without much input or intervention by higher levels of management and in that in a structure like that, you could still have like that walled garden of like, once you're a VP or higher, it's just like, we don't even want to look at you, you are plebes. And, and as long as the team is still Agile, they can still like that team could function effectively on its own in an Agile environment, even though there's like this weird, like strange hierarchical system once you get above the team, but I think for pretty much any company to reach its full potential. But for a team to function properly and in a way that's conducive to people doing their best work? Yes, you that sense of equality and openness.

Jacob Iwinski 1:06:33

When you see Agile lead or ScrumMaster on a job description, or LinkedIn profile, what does that tell you about that specific role and its duties?

FAANG Developer 1:06:50

If I actually saw a job title of ScrumMaster, I would personally think that that is a job that requires almost no actual work. Because anytime I've like I've never heard of someone's entire role just being like, Agile lead, unless it's like a massive company. That is the person who is involved in like, directing, and implementing, like Agile practices across every single level of the company and every team. Unless it was an upper-level management position, and like the title was Agile lead, I would think that somebody either had mislabeled a job, or that somebody was just kind of using buzzwords. Now, as far as like, line items go on a job if I saw like Scrum leader or Agile leader, whatever listed as like the line item for some other position, I would just assume that it's like, oh, this is probably like a management position like you're in charge of orchestrating the day to day and probably orchestrating the, the people or teams that that make up the company.

Jacob Iwinski 1:09:48

Are there any other significant benefits or drawbacks to the Agile Model as a whole that we did not already discuss otherwise?

FAANG Developer 1:10:00

I think the only one that comes to mind is that it can be a little bit more time consuming. Just in that, if you're following a very linear process when you are done, then you're done. Whereas if you're following Agile, you could end up going through various iterations, going back through steps, and that you can end up spending more time. But I think at the end of the day, you'll find that there are far fewer headaches. And far better results as a result of following an Agile process rather than one that's more linear.

Jacob Iwinski 1:10:49

Thank you. Last question. Question. 23. At the end of the day, do you feel there's anything special about Agile or Agile based project management?

FAANG Developer 1:11:04

Hmm. Yes and no.

So I would say yes, in that we've finally reached the point where, again, there's more of this equality between managers and the people working under them. I don't know if that's as much a result of people finding that Agile works better as it's just like, a wider change in, in workplace culture over the years. But I would also say no, it's not special, because I think pretty much everywhere has realized that this is a much better way, of doing things that is far more inclusive, and, and promotes equality. You will find some form of Agile, pretty much everywhere you go. So in that sense, it is no longer special, it's just kind of become like an accepted norm, at least at least in larger companies, especially any that are involved in tech. Because they found that the more flexible and iterative open model of doing things just makes more sense and leads to better outcomes at the end. I think if you look at it from a more like historical context, Agile is extremely special because for like literal 1000s of years, things were done in a very hierarchical and linear way. And it has only been in the past 20 to 30 years that that has changed.

Jacob Iwinski 1:13:05

To talk on that last point, what seems like common sense is only relevant because of the historical context. I guess, if we wanted to have an example or metaphor, kind of like a movie that does something special. And then every movie afterward copies. When you go back and watch the movie, you're like, this isn't special. And it's like, actually it is because it was the first one to do it.

FAANG Developer 1:13:49

Exactly. It's yeah, it's special. It's special from like, a historical context, because it's a very drastic departure from the way things just were always done. But at the same time, now, it has become so ubiquitous that it is it is no longer special. If you find an Agile environment, it's like oh, that's not really special. They're just doing it the way they should be.

Jacob Iwinski 1:14:21

Thank you so much. I really appreciate your time.

Appendix C

INTERVIEW TRANSCRIPT – FAANG PROJECT MANAGER, SCRUMMASTER

Jacob Iwinski 0:01

This meeting is recorded. Hello, I'm Jacob Iwinski. And I'm conducting this interview for the purposes of an honors thesis regarding Agile and Agile methodology.

Question one, please describe your role in current and past interactions with Agile.

FAANG Project Manager 0:24

I got into the tech space back in 97. So initially, I jumped in on the web design fields, when I had started. I created my own company back then. Small company, we had 20 to 30 employees, I ran that for about 15 years, I concentrated a lot on the project management side of things and training our project managers. And then eventually, I sold the company and went to work for enterprise. And then after about four or five years, I migrated on to (Startup Company) which got acquired by (FAANG Company). The first 15 years, when I was running things, it was almost always in a Waterfall type mentality. When I shifted over into Enterprise, that was my first real jump into learning the Agile space & getting Agile certifications. I took a company through the Agile transformation. And then when I went to (Startup Company), they had an Agile implementation but didn't fully understand. How do we scale it? How do we refine the processes? So each time it's been a little bit of a growth in the Agile? So specifically, I've been concentrating the Agile space for the last decade. So yeah, it's it's kind of been the first 15 and Waterfall in the next 10 and Agile, a little bit of the interaction.

Jacob Iwinski 2:11

In your own words, define Agile.

FAANG Project Manager 2:25

Yeah, a great place to start on that is the Agile Manifesto. So it's funny because it's a one or two page website that looks like it's out of the 90s, because it was out of the 90s. And it hasn't been updated intentionally because they don't want to update it. Because they want the heart of Agile, to remain pure. So I think it's very intentional the way that looks now. I view it is more of a mindset than anything, it's definitely not a framework. It's definitely not a toolset. So I view it much more into the mindset area, because they just it's boiled down to four very core beliefs. And then they only break down a gift from there is 12 principles. So if you just Google Agile Manifesto, you find that site and you can see those things. And the thing that jumps out there is it's not overly prescriptive. Like if you're if you're trying to say, am I Agile, you can read those statements and evaluate them at your company and say, you know, do these things apply to us? Or do we fall short in these areas? And these things tend to burn us so we don't do that. So yeah, it's a very open ended statement of when somebody says, Are you Agile? It's, it's not a cut and dry. We do this we do that it very much falls into day in day out, do you follow these, these beliefs as a mindset? And are you able to apply to work? And if you can say yes to those things? And yeah, you're probably on a pretty good, could it be a good place in terms of being Agile?

Jacob Iwinski 4:02

Thank you. Question three. What forms of Agile Do you have experience with as far as you know, Scrum Kanban, XP,

FAANG Project Manager 4:10

etc. Yeah, most Scrum and Kanban would be the two that I have the highest amount of experience in Scrum way more prevalently. I manage multiple teams and day at day in day out. We're utilizing Scrum. So that's where most of my knowledge will be.

Jacob Iwinski 4:31

Due, do you have any past relationship or experience with Lean or Six Sigma?

FAANG Project Manager 4:39

Not specific training, but a lot of the core of Agile is built off of lean. So it's kind of a yes or no question like no, I don't have formalized lean training. But if you look at if you go back to the actual lean processes, and you go through and look at like what did they define As you know, the eight lean practices and the I forget how many measures of voice they have in their in their philosophy. But if you look at and then you compare him to Agile, you can very much see that when they built Agile, they looked at that and said, yes, you're onto something. But now we need to evolve into a development space more. And they started to take those practices and just tweak them a little bit, but they're one in the same in the fact that Agile was really built off of a lean base.

Jacob Iwinski 5:34

Have you ever worked in a Waterfall environment, and can describe what that was like for you?

FAANG Project Manager 5:44

Yeah, so So Waterfall has a place. It works in very simple repeatable projects. So it especially if you have a client evolved it. It's not that it's better for it. But I think sometimes it's easier for a client to understand something when they feel like they have a fixed a fixed project where they feel like they know how much time it's going to take, they know how much money it's going to cost. So in a repeatable process, where you can go through and say, so, you know, back then it was, you know, we would go through, hey, we're in Discovery now or design. Now we're in development. Now we're in the discovery or in the delivery phase. So either, you can kind of cover those four and you create repeatable processes. Where it starts to fall apart is when you have complexities in there. So when there aren't things in the project that were expected that start getting thrown in? When you sandwich that deliverable in and you aren't negotiating anything out, you're expanding the timeline. And you're starting to throw things off. So what ends up happening is to make up for that people pad their projects, they put in these little buffers to try to assume all the extras that are going to get in. But the bigger your project gets in, the more complex it is the messier that becomes. So again, it works in certain environments. And then when you get into larger, more complex projects, that's very difficult to do. The other thing that ends up happening is you're making most or all of the decisions at the beginning of the project, right? Because you're writing a project plan up, then when do you know the least about your project at the beginning of the project. So you're going off of a plan that was written in a point of time when you know the absolute least about the project and have the most ill informed decisions. And then you feel a necessity to stay to it, right? Because it's your plan. You don't

want to go back to a client or something and be like, ah, actually, we pivoted off this entire thing. And now it's going to cost you a bunch more money. People stick to that plan, because they believe they have to. And then things that they discover midway, they either are going to cover up or say we'll get to it another time. So again, that's where the Waterfall starts to break apart is you know, when those things start to interject. So my experience was it works in certain environments can absolutely be horrible in other environments, depending on the type of project.

Jacob Iwinski 8:30

So if you had to create like a small but concrete list of the environments where Waterfall works, or is necessary, what would those be?

FAANG Project Manager 8:39

Simple repeatable processes. So simple, repeatable processes can show up in any in any industry. I mean, literally any industry, you can think of even mine, I will have occasionally projects that are rinse and repeats off of something we've already seen or done. In that case, I can just define the whole thing out front Waterfall and put it together. So I don't get a lot of those types of projects. But I mean, when they do crop up, there's no reason to make it overly complex but doesn't need to be.

Jacob Iwinski 9:12

So say like an academic institution wants to implement (Startup Company) product. Right? And you've already sold (product) at another academic institution at roughly the same size with roughly the same organizational structure. The second rollout can be a far more Waterfall style method?

FAANG Project Manager 9:39

That's a good example because the development of building (Startup Company) behind the scenes we very much had to be Agile with but you're right, the rollout. We've rolled this out to so many 1000s of customers at this point that no we don't want to be reinventing the wheel each time or have it I mean, it is a very repeatable process to put this into universities, hospitals. company, any company? And you're right, that would be a good example of a project plan that you just Waterfall it.

Jacob Iwinski 10:08

Question five, how was working it is working in an Agile environment, different from working in a Waterfall environment? Specifically you working under it as opposed to managing.

FAANG Project Manager 10:25

So there's, there's two things that always pop to mind in Agile that I think is feels a lot different when working in it. The first is you have autonomy to make your own decisions. So in Waterfall, that decision is kind of made for you, right? There's a project plan that's telling you what you have to do that that project plan came from somewhere that probably wasn't you. And your job is to put that in place. So, in Agile, it's a little different because you have the autonomy.

I am going to even reference Scrum because they go hand in hand. Scrum is a much more spelled out framework that people use because they want to become more Agile. They say okay, Agile

isn't telling me a lot. It tells me the mindset. But I don't know how to get to that mindset, right? I don't know how to day to day functionally implement that. Scrum is a framework that specifically builds on that mindset, and allows you to have a repeatable process day in day out to actually be more Agile. So in Scrum, you're, you're on a repeatable sprint cycle, commonly two weeks, but it can be one week, four weeks, I've seen different things in different places, the two weeks is the most common in the industry. So every two weeks, we're developing our sprint, we're deciding what we want to do what we think is best in that moment that we need to be doing to fulfill the vision of our product manager and deliver the most value to our customer. So that's autonomy. Like we get to decide we're not beholden to a project plan that says, you got to do this, now you got to do this negative, there's no you got to log your hours to get paid for it. Like whatever we are, we're deciding what our fate is going to be and what is best for the people around us.

Autonomy is as powerful thing to an engineer, as it is to a project manager. It's your job to manage your own destiny, having that recognized makes you feel valued, and it's exciting. The other piece to it beyond the autonomy side, is you get a much more sustainable pace of work.

So in a, in a Waterfall space you got even with repeatable project plans, when that stuff gets interjected in the middle, the client typically doesn't want to delay the project, right? So what ends up happening when you're not willing to negotiate anything out, or you're not willing to, you know, to add more. So you say "I gotta get the same amount of work done. Now, in just a shorter period of time." We do that times a bunch of different projects. And what is happening is you get these burnout cycles, where people really, really push to get something done, get something done. And then the burnout, and then the next project, you start slow, because you've already been on a previous burnout. And yeah, you do get these waves.

Scrum is really trying to help Agile get a sustainable pace. In fact, it's actually in the principles and looking at the language they use right now. I know it's in there in terms of in the 12 principles, they have something that specifically calls out sustainable pace. Agile processes promote sustainable development, the sponsors, developers and users should be able to maintain a constant pace indefinitely. So it's actually in Agile that concept. In Agile since you're, choosing what you're going to be working on, you're handling the things that come at you.

The idea is I want my team working at something that's completely sustainable indefinitely, where they could keep working, and never come to me and say I feel burned out. Because if they felt burnt out, it was because they chose that sprint themselves. So even if they do work extra hours, they very rarely are going to feel burnout. They themselves are the ones that raise their hand says I want to do this. So those two things, the autonomy and the sustainable pace alone have so much of a different feel in an Agile space versus a Waterfall space like retention. I mean those things impact your retention of your employees beyond the value that you can put in to the actual customers and what you're going to get done.

Jacob Iwinski 15:04

Do you feel that Agile from a worker perspective, empowers equality and more transparency and ease of access between workers and management or senior management in like an ease of communication way that might not have been present? Prior to that in the corporate world?

FAANG Project Manager 15:34

With development there has to be a trust between the management level and the team level at that point. And there has to be an interaction because management now wants to know what's going on and what decisions are being made. So there's a communication there, that's two way in Waterfall is one way, usually, management saying this needs to be done. And then that's it, you're done with the communication aspect you've been given your thing to do, you do it. In an Agile space, there's going to be communication back through management, as new decisions need to be made, as we're seeing things. I'm in the middle of quarter right now, or mid quarter reprioritizing, what we plan on doing, and that's all funneling back through leadership, to help them understand what needs to be made and what we didn't expect a month and a half ago. So yeah, I think it definitely promotes a back and forth conversation that doesn't exist the other way.

Jacob Iwinski 16:48

Thank you. Question six. What did you think of Agile while you were worker? Compared to now that you're in management? How has that changed your perspective on Agile?

FAANG Project Manager 17:03

Yeah, it's, it's interesting, because yeah, when I was in the, because when I when I say project management, I don't consider myself a manager, like I could sort of manager as a manager of people. So when I was manager, you know. We were all in Waterfall mentality. And I don't know if I had discovered back then, because we weren't using the Agile, I didn't know what I didn't know, like, I kind of you do your best to, to empower people. But I look back, and I really don't think I was. Because everything was so top down on how we approach things. If I look back, it would almost be like, if I knew what I knew now, and could have implemented Agile back then. I think we would just been happier. I mean, not that we didn't have happy employees. But I think things would have gone a lot better in terms of people feel empowered around us too. Because I don't want to be telling people what to do. You hire people, really smart people, so they can tell you what to do. So you're basically allowing that to happen. These are very, very smart people that we hired for a very good reason. It is a waste of my time to be trying to tell them how to do their jobs. I think it would have changed things quite a bit. If we would approach it back then like that.

Jacob Iwinski 18:39

So while having a title of, you know, manager, it basically gets kind of to the more, you know, American centric idea of like empowerment management, right? Where you are there to provide the resources to help workers, individual contributors achieve their personal best, as opposed to, you know, lay down the law as it might have been before.

FAANG Project Manager 19:05

Yeah, or manage hours or make sure they're doing work or any of that stuff.

Jacob Iwinski 19:10

Thank you. Question seven, what kind of Agile training have you received? And has it changed the way you approached the topic and management in general?

FAANG Project Manager 19:23

Yeah, so I've done a handful of different things. It's in terms of Scrum specific trainings. I've let's see, there's three levels through this Certified ScrumMaster. Branch. So you've got your CSM, which is your Certified ScrumMaster. You've got your ACSM which is your Advanced Certified ScrumMaster. And then you've got a whole bunch of them except the letters on this. Your GS what is it? You're CSPSM, I have to look that one up. It's in my LinkedIn profile, you can get it there, it's. So there's three levels that you can go through over time. I've been through all three of those. I've also there's, so there's so many frameworks out there. And they're being taught by different places, too. So it's not like always just one thing. scaled. Let's see, so SAFE, which is a Scaled Agile Framework. So I'm safe certified as well, which is, that's basically a framework that's attempting to say, Okay, you're, you're trying to be Agile, you're likely doing Scrum. But now you're trying to scale it in so many teams across the org that need to be aligned all at once they developed a Scaled Agile Framework to try to answer or how to actually have these communications all across all these teams in real time and actually still be Agile. We don't specifically use that currently at (Startup Company), we did it more back at the place. But I am at least certified. And to answer that question. I might have to look up because it's been like 720 something years of doing this. I know I've gotten other ones along the way. But I'd have to look for those are probably if somebody's training in my industry, I'd at least be looking for stuff like the CSMs and stuff on that track.

Jacob Iwinski 21:28

Question Eight, can you speak to the resources that were available to you within your training.

FAANG Project Manager 21:35

So most of them are instructor led. And then they have Course Guides. The interesting thing is, in all the things I've done, the approach is almost they're trying to be a little Agile with the approach. So they're actually trying to have self organizing teams within training. So a lot of times, it's not, here's a PowerPoint on the speak for you for two days straight. And then have you do this, a lot of times they're breaking you off. And your teams are self organizing, taking the topics that they have sharing experiences, trying to come up with stuff. And then the instructors are there kind of guiding along the ways, almost like a ScrumMaster would. So I've seen that be very prevalent in the trainings of how they've approached it. So usually as resources, have, you know, the instructor and materials and your classmates who are all trying to collaboratively work together throughout the training to evolve what people know.

Jacob Iwinski 22:34

Thank you. Were there Question Nine, Were there ever any peer reviewed or like scientific resources available in those trainings?

FAANG Project Manager 22:46

So it's interesting, because I've seen a lot of that stuff. I don't know if I buy it, because very rarely will you have can you have a truly like a project that you can't do a project both in Waterfall and Agile, and at the same time, then report back results. And there's so many variables that can change by project, to really try to put a number around it, I have seen that type of stuff done. The other thing, too, is I think anytime that you see a number like that, I wonder there's always an agenda of who put it out there. If you're doing the study did it for a reason that

you're either endorsing Agile, you're endorsing the scale term, if you're endorsing something where you want the numbers to show something. So this might just be me and my personality is I tend to throw a lot of that out. And I go off a feel. It's like I've been doing this long enough to know if this feels like we're at a better pace, if we're delivering better if we're getting a better end product. Those are the things I can usually have a pretty good feel for. And I can say without a doubt. I mean, the reason I haven't shifted when I went to Agile 10 years ago is because it works a heck of a lot better for us.

Jacob Iwinski 24:02

Question 10 Is there ever a lack of consensus on what Agile means or meant in your organization's or training?

FAANG Project Manager 24:15

Yeah, absolutely. I've been fortunate that my last two companies have been really good about this. So I see it probably more outside my own companies and industry more. But it still happens even within my own company. The one I see most commonly, starting with the worst case scenario is when people have really no idea what Agile means. I think they mean they think it means be flexible. Like literally like the exact like definition of Agile is "your agility" and "you can move around." They're trying to make up a definition of what Agile is without having any kind of training or understanding what Agile is. The other one I see as with Scrum, specifically is people put in like a daily connect, and be like, yeah, we're Scrum - we do a daily connect, or like, we're Scrum - we meet and we talk about what the Sprint's will look like, and it's like, there, there is so much more than just being doing Scrum that has nothing to do with the fact that you've scheduled a couple of meetings. That's not the essence of what they're after. It's the same thing with Agile, it's like, just because you feel like you can think flexible or give your employees flexibility. And, you know, just because you put them in a couple pods together, so they could collaborate that didn't suddenly make them Agile. So I think it just comes from a lack of understanding of never really understanding Agile in the first place without evaluating, are we doing this correctly or not?

Jacob Iwinski 25:52

So to build on that, I guess, any company can organize daily stand ups, or sprint debriefs or claim they work in a sprint environment. But the truly Agile part requires the worker, the individual contributor lead empowerment component, and without that, you're not Agile.

FAANG Project Manager 26:16

Yeah, let's say you had a company that was running daily connects, and let's say they're running, sprint planning, let's leave it at that and nothing else. Go through the 12 principles, just measure this alone, this is a checklist. Just start picking some out. I'm just picking randomly from the list like, one of them, one of the 12 principles in Agile is working software as a primary measure of progress. So when you measure your progress throughout the spread, where are you concentrating? And how many hours are story points you got done? Or was your measure truly working software? Do I have anything working? Do I have anything? You know, start picking other things. Simplicity, the art of maximizing the amount of work not done is essential? Do we say no to anything throughout this process? Where are we strategic? And how are we picking this stuff up? Pick another one at regular intervals, the team reflects on how to become more

effective than tunes and adjust its behavior accordingly. Would you do in the last two weeks that was different than the two weeks before that? Did we have any reflection or introspection on how we were accomplishing things to find any better method. So you kind of do that, where you just go through all 12. Like, forget what meetings you're running? Am I doing these things, and if I am, regardless of the meetings, I'm in good shape.

But I could run every Scrum meeting under the sun, I could be doing backlog grooming, you know. But if you can't look at those things in the list and the Agile principles and nod your head and say, Yeah, I think we're pretty good or on our way on those things, then no, I think there's probably a pretty good big gap as far as are you actually being Agile or not.

Jacob Iwinski 27:58

I guess that really gets into Agile, agility, agility as a mindset as opposed to a set of meetings or tools.

FAANG Project Manager 28:07

Absolutely, yeah, Scrum is there to try to facilitate or help in that process of being more Agile. But Scrum is not in and of itself Agile. If you just do Scrum, and don't understand the essence of Agile, you'd still not be Agile at all.

Jacob Iwinski 28:23

Do you have experience or you know, first or second hand experience with companies that operate in that kind of environment?

FAANG Project Manager 28:31

I'll take it a step further. I can be the cause of it sometimes. Like, it has nothing to do often with the company itself. It's like I can slip myself like I could get into a habit on how we're doing things or the way we're doing meetings, and then slip back and start to look at that list again and realize, you know, I haven't been paying attention much to this or, you know, start picking other ones out here.

Let's just go into the list. If I got into a big release that we were putting together and then look back, it's like, well, could we have chunked that up? Could we have released those into five mini releases versus putting them into one, for example, like, those are the types of things where like, I've been doing this for a long time, I might have the best intentions. And I'm trained in this. But if I'm not looking at these things and pushing myself to make sure, I can be the cause of my own team, just getting into a habit and getting away from what the core principles are.

There are times companies can do it to. Or it's management that's doing it like you know, the one I think that's most common in companies, is they want you to be Agile, but then at the same time They want the outputs that you give them to all be back in the old school format (Waterfall). So for example, you're working on a project, you don't want to find all the requirements upfront, you want the ability to make these decisions in real time. But management comes back and says, I still want you to tell me exactly how long it's going to take exactly how much money it's going to cost. But if they keep pushing back and saying, but give me a number and try to hold you to it, they themselves aren't allowing you to truly be Agile. And if you are, you're being Agile in very

small pockets that are safe. Knowing that you still have to be in there, their boundaries. So I think that's the other ones you see is the conflict back and forth between management and worker of if they don't truly believe it themselves, and allow you to work in that space. You know, people are gonna be banging their heads against the wall at some point.

Jacob Iwinski 31:03

Thank you. How do teams and companies negotiate contracts or fixed contracts for services or products while still working in an Agile way?

FAANG Project Manager 31:21

Yeah, I probably have the least amount of good experience to talk to you about, only because I've worked in software development, where the company, our internal company, basically, is my initial stakeholder. So I'm not writing a contract, it's different than agency work.

All I can say is, in agency work, you are writing a lot more contracts was living for. Even though we sell (Product), we're not making (Product) specifically for one client, right? The clients that they can influence us in terms of the features and stuff like that we're constantly seeking out opinions for that type of stuff. But, you know, if I'm working for Penn State, for example, they're not telling me what's on their roadmap, and when I must deliver it. Having a contract for when the new feature comes out, or whatever it is not like that. You do get that though, with some companies that have development goals under contract.

I have less experience in that area of how Agile can be deployed into it. I know that there's been many, many attempts, like if you, if you search for Agency Agile, that's actually a framework, There are people who teach Agency Agile to agencies, and try to help them blend in Agile with writing contracts for customers. You can make the customer, your product owner on the team, and actually do crazy things about like that, so there are things out there that attempt to blend Agile in with a customer who has very real world demands, but I still need to know how much it will cost and when. I haven't been in that world. I either worked on the customer side and completely did a Waterfall, or I skipped over to software development where we were our own internal customer. And I've been in that space running Agile forever. And in a way I have it easier because I don't have that complication of an outside factor that I'm holding on a contractor.

Jacob Iwinski 33:39

Agile claims to significantly improve customer satisfaction and productivity. But this requires the customers to participate or the stakeholders to participate in the Agile process. Do customers ever not want to do that? What happens when there's tension between what teams and want? Can you just talk a little bit about that?

FAANG Project Manager 34:08

I don't have as much direct experience with that. My world is a little different, again we were not developing for a specific customer, but we still need to involve the customers in our decision. A lot of times what we're going to be doing user research studies, where we're actually interviewing customers about hypotheticals, or we're showing them product and actually having them interact with it and finding where they're stumbling. We're seeking their opinions, but we're not beholden to a contract to actually deliver any specific thing with them. So the relationship's a little

different, they're part of our process, but they're not really part of the direct Agile process, like in Agency Agile.

But in our case, it still very much influences what we're going to be doing. I mean, we've had ones before, like, I'll give an example. And I'll try to distract us a little bit. We're rolling out one of our new products within (Startup Company), and the one of so we use something called OKRs objectives, key results for every quarter. And one of our keys was we wanted a certain delivery, like we wanted a certain amount of adoption, like a percentage of customers who have converted over from this version to this version. We didn't hit the number. Not even close actually. And so what we did was, again, rather than just keep following down a Waterfall process and keep on going, we hit the brakes for a second and said, Okay, something happened here, something we got wrong early on in our guesstimate of how this is going to work. And what we found was there was something on the client side, that would not enable them to even make the transition. It wasn't their fault. They just weren't ever going to be able to make that switch over until they had either they got a completely different method, or they had some circumstance change. We didn't know that or didn't anticipate that at the beginning. So we basically rewrote the whole thing, like the project plan, just evolved it and then on the spot. So we flipped everything around changed how we were slotting things. The customer was a huge influence on that. They are the ones that helped us understand why adoption wasn't going to work the way we thought it would work in our heads. So they were part of that process without knowing they were part of the process. So you can't ignore them as part of it.

Jacob Iwinski 37:08

No, but that's definitely Agile thinking. In the interest of attaining a gold standard hybrid model between, you know, Waterfall, Scrum Kanban, you know, what specific methods or behaviors do you feel are most significant in helping to deliver value?

FAANG Project Manager 38:08

Yeah, so the thing I would say for that is it's less about how you start and more about how you evolve. So we're going to take our best stab at what we think we need to get to get going. And then I'm going to rely on retrospectives every single sprint cycle every single two weeks.

At the very end of that Sprint's as we're closing it out, we're going to hold a retrospective as a team, we're going to start to talk through things. And what I'm hoping for is that the gaps in the things that aren't going well are going to spell out our next steps and whatever we need to change whether it's hybrid in certain models, taking something from somewhere else. I want the team to help me figure that out, come to that conclusion, come up with ideas, and then start to make evolutions.

Because like, I manage multiple teams, and I don't manage them the same way. I have teams working within the engineering space a (Startup Company) that have completely different processes. And it's not because it's any easier on me to run two different processes for different teams. Say this process tends to work better for this other team and the certain things they do. A practical example is I have both of my teams do story pointing for sizing the task, that's common in Scrum. One of my teams decided they wanted to completely stop story pointing and believe they could still make it work with no estimation basically. We ran a two quarter experiment on it

where we did everything in absence of estimating, and actually found that the team that didn't do estimation ended up being more accurate with their deliverables than the team that does estimation. So it is something like that, where it's hybrid and multiple approaches. So that was a piece that team grabbed wanted to experiment with, and then found it worked for them, and then made that part of their process long term. So it's that type of thing from where the hybrids come, not from me necessarily planning them out, but more of just a team, adapting to the things that we're seeing around us, being reflective on it, and then just saying, let's try and experiment and if it works, awesome, if it doesn't fail fast, and what does revert back or do something else and come up with another idea.

Jacob Iwinski 40:56

You've got to be Agile in your implementation of Agile. Has increased customer participation or rapid prototyping ever resulted in reduced productivity or maybe rushed designs?

FAANG Project Manager 41:19

Yeah, it goes back into more of the agency thing. I can give you one example. Because sometimes you don't need a client to have pressure, right? Sometimes the pressure comes from your competitors, putting something out there, and then realizing you're behind. So it's like you're developing something, they develop it faster, and then actually get out there with like a feature. You're like, oh, I don't even think about and you're like, oh, shoot, I'm behind on this.

So you can get that pressure, even without a customer. But yeah, it's at that point when you when the most common reaction you get from management is to throw more people at it, right? But a team never develops into a self organizing team because they're thrust together quickly. You also run into issues with the volume of people that aren't necessarily within the recommendation size. I think Scrum recommends somewhere between five to eight developers at most.

Jacob Iwinski 42:31

The Pizza Size?

FAANG Project Manager 42:33

Yeah. You know, so when I end up with a team with 10, developers, or more, people are just trying to crank through because we feel like we're behind. That will make it harder to run Scrum. Because Scrum came up with those boundaries for a reason, because they started to find that things started to break when we started to put more people than you should on teams. At that point you should have more teams. And when you start to break down Scrum, then you start to not be as Agile as you want to be. And then that's when the team starts not having a good sustainable pace and it starts to feel out of whack. Things start breaking. So yeah, absolutely, it can happen, even without a customer.

Jacob Iwinski 43:30

What was the transition to Agile like, in the programming and project management space both for you, and if you saw that transformation, through the industry as a whole?

FAANG Project Manager 43:54

Yeah, so the thing that if I were speaking to like younger self back when I was pre-Agile, the thing I would have told myself is stop being the in-between to everything. I think project managers tried to view themselves in traditional project management as the gatekeepers to everything, right? They want to protect their team, so they will limit somebody from the outside trying to interact with the team. I'll protect my engineers. Same thing, if they want to talk back, tell it to me, I'll go tell it to management. There needs to be an approval of the direction change. Send it through me the project manager, I'll bring it up.

I think there's very much traditional project management, kind of empower the project manager. I think it's backwards because what you end up doing is devaluing the why to the team, like, they never understand the why, because they were never part of the process to come up with these decisions. So it's like, if an engineer in Agile wants to talk to somebody on another team? Don't tell the project manager, you can give him an update later, just talk. And if management wants to come talk to our team and explain something, they can just explain it, you don't need to talk to me first.

I didn't realize that back in the day. It was always my instinct was protect the team, but when you when you stop protecting the team, and let them just speak, make decisions, and let them go talk to people, it's amazing. The end result is not them burning out. If they feel like they have too many people all around, we can figure it out in a retro if we need to. But this makes them feel empowered, that they get to figure these things out, they get to talk to management and shape beliefs, and you know, articulate the message. And if the message comes back to them, it's probably going to come through a lot better for the people making the decision than it is through me. So that's the biggest thing that jumped out as me and the transitions of what it meant to be in project management. I stopped making myself so central to the process. And it freed me up to do more of my job and project management and less of trying to be an in-between to everybody, which is not adding value. Me being in between does not speed the team up at all. So it's like I'm doing this exercise to try to do something that isn't helping anybody.

FAANG Project Manager 48:30

That scenario, when you're told, you know, basically stay in your lane and funnel it through these people? Do you feel empowered, or be creative at all in that scenario? Do you like when people are put in lanes to limit how they do things.

The other effect is siloing. Typically, teams that run in Waterfall, tend to hire more in silos intentionally because they want it to be a little more like an assembly line. Like, this person does this, this person does this, this person does this, hands it off over here, communicates back, etc.

If you have a team like that, though, what happens when person six in that process suddenly is on PTO unexpectedly for two weeks from an illness? The whole thing breaks down, right? Like, you know, you don't get a backup, you can't bring somebody up to speed quickly enough. Nobody knows what that person's work was. They probably didn't know they're even going to be out so nobody else even has a handoff. So we just basically broke the machine. In an Agile, typically, self-organizing teams tend to do things like pair programming, XP was big early on with pair programming, which from that adopts a lot of these, methodologies that play off each other. I've never got any XP training, but we use elements of XP every day. So typically on an

Agile team, you'll see a lot of pair programming, a lot of people doing code reviews of each other's work. So the idea is, if I have somebody out for two weeks, the team just picks them up, like somebody else would have known what they're working on would have already seen code reviews would have already been in the same planning meetings would have already done a peer review session at that point. One of the big things is, "Let's have each other's backs." If a person is out for the next two weeks, we still have goals. How are we going to do that? We're all going to work together, make sure it gets out, do a demo of it working at the end of the sprint, talk about it during the retro and how we performed on it, see if there's a better way that we could be handling this as a group.

I mean, that's like night and day difference from, we broke the machine/we don't know what to do. And that's to me, like when you say you're running an Agile mindset, if somebody can give me that example, and show and walk me through that process of how they just adapted the whole thing and made it work. That's being Agile, not running a Scrum meeting or, you know, whatever else people come up with, to think they're Agile.

Jacob Iwinski 51:37

To use a really nerdy example, being Agile with XP, and Scrum and pair programming. With self-organizing teams is like project managing in RAID five, where you can have redundancy built into the system, and you can have disks go offline, and you can slot disks in and out and replace things without losing where you were.

FAANG Project Manager 52:09

Absolutely, yeah. So you get all these extra effects that you are looking for with Agile.

Jacob Iwinski 52:33

Has as Agile has shifted from just kind of programming and into a far broader overarching project management for pretty much every industry? Can you talk a little bit about that and how you've seen Agile impact the rest of the corporate world?

FAANG Project Manager 53:04

Yeah, it's um, so there's a conference I go to yearly that's called *Agile and Beyond*. So they actually hold it local to us. There's a lot of Agile competence all throughout the world. It's not just all people in my industry. So when they say *Agile and Beyond*, it's people that are utilizing Agile or want to utilize it better. The industries that are there? It's everything. I mean, you get financial companies, you get automotive companies, educational companies, it's not just a bunch of rinse and repeat of people in the tech space. And it's interesting hearing the stories because the applications of Agile, when abstracted are all very, very similar. But their circumstances are very much different. So you see a company like Ford, who's been hugely invested in the Agile over the last several years, and how they're redesigning how they develop cars. And everything internally is based around Agile methodologies.

It's really interesting stuff because you start realizing that even the intent of the founders of Agile, I don't know if they fully understood how far that would end up going. And how many applications that are ended up having. Yeah, I think it's not a silver bullet to anybody. And Not all companies who do run it are 100% successful for various reasons. I think that the companies

who have done well with it are the ones who are very intentional about it, who understand it before they make the decision.

So one thing you'll look at is when you do an Agile transformation, this can be any industry. If you're hiring an outside, let's say you hire an outside person, and say, we don't know enough, we need you to come in to help us run our transformation to agility. You know, the first thing somebody's going to tell you is, you know, you're in this for multiple years, right? You understand, like, this is not something where in two months, you're going to be looking back and be like, it's all flowing, it's all Agile, my customers getting value, like, you know, that mind shift set is so ingrained in companies that the ones that do it, I mean, one of the first things they got to realize is, you know, this is a multi-year journey for you guys to do this, and you're going to have some many wins along the way, you're probably going to have some setbacks along the way. But the end result that you want is when you look at this, after a year or two years more, you do look back, and you say, it's radically changed our company and how we approach things.

Where we are now is better and customers are getting more value out of what we're doing. So that's the shift I've seen in the corporate world for those who have worked for. You hear a lot of amazing stories from some pretty large companies that have been successful in it, and then you've got a mix of the ones that either never understood it or just weren't patient enough to ever invest correctly into actually making it happen. And you hear those as well. But either way, it's had a pretty big reach into the corporate world where at this point, if you if you talk to somebody about Agile, most likely you're not going to get a blank look. People know what you're talking about, have experienced it on some level and have some thought onto it.

Jacob Iwinski 56:51

Agile is not just for individual teams, but also all the way up and down, like a corporate hierarchy, how do a manager's manager's become Agile?

FAANG Project Manager 57:16

Yeah, it's so the Scaled Agile Framework is interesting, because it attempted to really put a lot of process behind it. You have these big pie planning sessions, which stands for program increment. So it's basically massive quarterly planning that involve all the teams and it's interesting, when I ran it, I didn't dislike it, but it I don't miss it either. When scaling Agile, I think it's less about process and more about better communication. Before you start scaling Agile are the teams that you are attempting to scale with at the base? Are they Agile? Because what I don't want to do is take a bunch of mediocre to bad Agile teams and scale that.

That just becomes chaos. So the first thing we look at is do my teams have this right in the first place? One team, two teams, five teams, whatever it is. I can scale off that now. So I can start to scale them, start to have them preach the processes into the other teams. And then start working to figure out how do we best communicate? What's the cadences? What are the types of meetings? Who needs to be involved?

It may not be a one size fits all for, let's say, the 20 teams that you're trying to do this for? It might be that certain teams interact with each other in a certain way. So usually, in the scaling process, that's what I'm looking for is, is can I take some really good performing teams start the

build out, have them teach each other and learn from each other? And then really focus on how they do communication because, it might just be that they do *Scrum of Scrums* with each other, which is a popular technique for how you involve multiple teams together. It might be a little more asynchronous where they have Slack channels built for team-to-team communication that they can throw things over the wall back and forth. Communication. It's not that there's a right or wrong way. It's about intentionality. If you're intentional about how you do those communications and know how you're going to handle it. I think you have a shot then in starting to expand the volume of teams and have them actually know what's going on.

The part that I think you can put some lightweight processes in that doesn't become too overbearing is when you do something soft like setting a standard for making sure, you're always on the same sprint cycle. That's a given. Maybe you say okay, because you're on the same sprint cycle. We're all working on the same quarters as a company. For quarter one, I have an expectation that all teams are going to submit their OKRs in draft version by this date, so that everybody can review them and make sure that there's no gotchas. You can put in lightweight processes and stuff like that. But I don't think you can, from top down, dictate all the things that every team is going to do at all moments, because then we have another anti-pattern and Agile, where we're telling people what to do, and not allowing them to be self-organized. And so it's this weird kind of mix of like, I need something to bring us together. But I also need to trust that the people that are self-organizing and going to fill in some of those gaps for me as we go.

Jacob Iwinski 1:00:24

Then could you talk a little bit about kind of like the Agile transformation? Because it seems, oftentimes, it's higher up management or a CEO who says "I want to make our organization Agile." But it seems that it's also important to start that transformation, and build it from the bottom up and make sure starting with the teams, and then their managers, that way everyone actually has solid foundations with Agile as you move up the chain. So how did your experience with a transition to Agile play out? And what are the challenges that can come along with that?

FAANG Project Manager 1:01:03

So yes, to both of those things, management should be getting trained. And the teams down there should be getting trained as well. There are different trainings. So the team can get trained in Agile. If you start putting ScrumMasters, product owners and other roles in place, they can get trained and have certifications. There's leadership training for Agile, specifically meant for C levels, there's one's meant for mid-levels it, the idea is, everybody needs to become part of that process. Because we're going to start falling apart if the team gets trained and Agile, but the management doesn't support them or understand what the teams need. If management gets trained, and never bothers to get the actual workers on board, it's going to fall apart. As far as transformations go, typically, if it's somebody has never done it, it's hard for them to even know where to start. Like "What training do I send people to? What's the expectations?" You may see them either bring in consultants who have experienced this, or they may hire on their own C level. Somebody who has experience in this so they might, you know, take a CTO and say, okay, as Chief Technology Officer, do you have experience taking these other companies Agile? We need you to guide us through. Like, what do you need to build the teams? What do you need the rest of us C level people to be doing? What do we need to do? What are the expectations we should have? And that person may guide them through and say, okay, you know, this is going to

be a multi-year journey, I need to start with these teams. Currently, you have project managers who might become ScrumMasters, you have BAs (Business Analysts), who might be product owners, or maybe I have gaps, and I don't have roles that I even need, and I need you guys to open up.

FAANG Project Manager 1:02:53

So just an example, when you make a transformation, it may not be one to one, right? Developers and engineers are more of one to one, but like the roles around it? You may not have a product person. In your traditional company, you may have business analysts, you may not have a ScrumMaster, you may have someone who's used to being a program manager, a project manager that that's different than the ScrumMaster. So like, when you're going through a transformation, there's a lot in there to unpack that somebody's going to have to guide a company through. Because if you just start switching people's roles, without being very intelligent about it, or knowing if those people even fit to those neutrals, it's going to be chaos, it's not going to work.

But if somebody's been through this, again, it's going to be a slow process where they might explain the whys, go through and make it very known everyone, this is a large scale transformation that's going to have benefit hopefully, to everybody, when we're done. We're going to interview everyone, talk to them about the most applicable transitions that can make, then help them through those and get them trainings. If it doesn't work for you, we'll figure something out. If we have gaps, we're going to open up racks, we're going to bring in new people that have experience in these areas that complement people. That's a very intentional approach to how you transition a company.

Companies that do that have a chance to be unsuccessful. Again, you can see the gaps of if a company doesn't do that, doesn't have the right expectation, doesn't have the patience, doesn't have someone that knows the steps to bring this through, and just start jamming things around and saying we're Agile so we're going to do this suddenly. It can be very painful to some companies.

Jacob Iwinski 1:04:30

Just a few more questions. When you see Agile lead or you know ScrumMaster, for example on a job description. What does that tell you about the role or what would you assume about that role or that person's job?

FAANG Project Manager 1:04:46

Yeah, titles are sometimes confused in our industry, because we're not always universal about them. I'm called a senior project manager. I do more Scrum. I'm more of a ScrumMaster, than I am a project manager. So Agile lead / ScrumMaster more of probably what I'd be looking for if I were to want to switch jobs. It would tell me is that they have an implementation of Scrum, or Agile, because they are bothering put the title on it.

Most likely in that role, I have dedicated teams to be working with. Hopefully they run in a formation. That's true to what we're trying to do with Scrum. Hopefully, they are self-organizing, hopefully, they can cross collaborate and work together. Hopefully, they are someone that they're going to find eight engineers, hopefully, they do have a product owner who I'm going to be

collaborating with who's going to be actually giving me priority so that I can help the team organize and make decisions with like, those are, those are expectations that if I saw a job description, I would hope I would see bolded out in some fashion, that these types of things exist. So then I would have shot as a ScrumMaster of actually coming in and being successful.

Jacob Iwinski 1:06:05

Thank you. Describe your day to day job and role as a ScrumMaster.

FAANG Project Manager 1:06:16

Yeah, so two teams tends to be a sweet spot for us. At three teams under management, you're basically a professional meeting attender at that point, because you're just bouncing around. With only one team running in good formation could almost be a too light, where you're always getting bored. Then you want to get into it some more with a different team or compare teams and run experiments back and forth. So we found to be a sweet spot. So, I have two teams.

FAANG Project Manager 1:06:55

So for a given team, I break things down in a time period. So it's like we're in q4, right now, where it's offset from calendar. So we're at the end of our fiscal year. So q4, like every quarter has six sprints. And then each of those six sprints has typically 10 working days in it if you don't have holidays, and things like that. So I'm breaking things down to a point where I have, I've worked with my team on a quarterly level to figure out what our OKRs are, we've we're actually going sprint by sprint, and then building out what we can actually accomplish first spreads. And then funneling down to, if I've done everything right, with a team, hopefully, on a day to day basis, they're able to just concentrate on what they're supposed to be doing that day. And just knock it out of the park at a sustainable pace. And know that we'll get to the other stuff tomorrow that we have planned or reevaluate as necessary.

That's a lot of my role then, just making sure that that flow is happening, so that every single day, the team is collaborating, solving problems, and getting past impediments. How I do that is a little bit open ended, but just like they have autonomy to pick the work, I have the autonomy to figure out the methods that are going to work for what I want to propose back to the team. But the key for me is, I want it to be much more facilitation than dictatorship. I never want to tell the team what they have to do. I'm there to facilitate conversation.

I'm hoping that they can come up with the solutions and how they're going to implement and I'll just be there again, helping making sure that that actually flows and works. And you know, if they get stuck, I can provide helpful in ways that we can become unstuck. You know, so really, on a day-to-day basis, I'm prepping for my daily connects and prepping for my upcoming sprint plans. If I'm at the start or end of a quarter there'd be activities I'm doing. There's also a handful of things that I will always be doing as a ScrumMaster.

The rest of the time, though, I'm deciding my tasks based on what's happening with the team. Maybe there was an experiment from last retro that they want to implement. And I've got to put a little brainpower on how we're going to actually take their ideas and bring them to life next sprint, and I might spend a little bit of time doing that, or you know, maybe there's a different tool out there. Maybe we've been using, *Mural* for our retros and I want to switch it up and bring

the retro into a different tool, maybe the format of the retro is stale and people are kind of getting bored during retros. I could change up the entire format for what the rest even looks like and the questions we are asking. Those are the types of things that, I'm just looking at throughout every single day, with the sole intention of everything I do is about making the engineers be able to be more Agile and deliver better.

Jacob Iwinski 1:10:18

So I was an RA at Penn State for two years. And the big thing was, you don't solve their problems, right. But you acquire that institutional knowledge of all the resources available. So that whenever someone does have a problem, you know exactly where to point them, so they can solve it themselves.

FAANG Project Manager 1:10:41

In some cases, yes. In other cases, also recognizing that they just may be the smarter person in the room than me, and that they're not solving it so that they can learn the lesson they're solving because I literally don't know the answer. And my job is to get the answer. And they know it. So like, it's not even always that mind trick of like, I want them to be able to fish themselves, you know, to thing like, I truly may not know the answer. I'm just going to be facilitating that road they could use to get there, hopefully.

Jacob Iwinski 1:11:19

Thank you. Just like three more questions, one. Are there any other significant benefits or drawbacks to the Agile Model as a whole that we have not already discussed?

FAANG Project Manager 1:11:37

In terms of drawbacks, it's interesting, because that's such a company-by-company dependent thing.

That I think drawbacks are probably very specific to the company itself. I think if you have a true deep understanding of what all these different methodologies are, it's not so much of a drawback as you may make a different decision to not do something like you may have a project where you're like, yeah, it doesn't make sense to be an Agile on this specific project. So it's not so much I view that as a drawback of Agile versus more of the responsibility of the person or people who are on the project, to just make an informed decision that we know what Agile is, it may not be the best thing for this particular project, just decide not to be that for this project and go into a different methodology. So yeah, just do it less about drawbacks and more about just put it back in the people have, it's their decision whether you want to put in place, it's the same thing with Agile in general, it's like, you don't have to do Scrum by the book Exactly. Like if something works better for your team or your company, make that decision. Just be intentional about it, make sure that you have some way of measuring your benefits in satisfaction. Measuring doesn't have to be purely empirical numbers. Have some way of understanding that this thing actually works. And if you're doing that, yeah, and it's less about drawbacks, and more about just better decision making.

Jacob Iwinski 1:13:05

Thank you. At the end of the day, do you feel that there's anything special about Agile based project management?

FAANG Project Manager 1:13:17

I don't know. Special? So I know we covered a lot of things in terms of how I think it's a lot different. And again, if I if I were talking to younger me, I just think there's so many things I didn't know at the time until I got into Agile. So I mean, I think that's what's special about it is just a lot of the things we talked about with you know, it's special, if you can make people feel empowered, to make their own decisions. It's special, if you can have engineers want to stay at the company in multiple years, and not feel like their job hunting, because they're burned out like that. That is special. I don't know. I mean, it's just an outcome of hopefully, if you're doing good, Agile, you know, it's special. If you can deliver something to a customer that they actually need to use, and actually knock it out of the park. It was because of the example of where we pivoted mid mid quarter on something because we realized this was just the wrong approach and that our customers weren't going to be able to use this the way we thought. I mean, in a way that special that we're going to be able to solve problems in a better way than if we were to just jam forward or the project plan because that's the way we thought we should be doing it. So I don't know, especially with the word, but that that I guess always special if you do those things.

Jacob Iwinski 1:14:42

Thank you so much. Participant 2.

BIBLIOGRAPHY

- (2001). Manifesto for Agile Software Development. <https://Agilemanifesto.org>
- Anthony, S. D., Patrick Viguerie, S., Schwartz, E. I., & Landeghem, J. van. (2018). 2018 *Corporate Longevity Forecast: Creative Destruction is Accelerating*.
www.innosight.com
- Bain, M. (2018). *High Performing Corporate Communication Teams*.
<https://www.researchgate.net/publication/331024001>
- Carvalho, B. V. de, & Mello, C. H. P. (2011). Scrum Agile product development method – literature review, analysis and classification. *Product Management & Development*, 9(1), 39–49. <https://doi.org/10.4322/pmd.2011.005>
- Developer, F. (2022, June 28). A Worker's Life With Agile. (J. Iwinski, Interviewer)
- Digite. (2021, October 25). *What is a Kanban board? - The fundamentals*.
<https://www.digite.com/kanban/kanban-board/>
- Dühring, L., & Zerfass, A. (2021). The Triple Role of Communications in Agile Organizations. *International Journal of Strategic Communication*, 15(2), 93–112.
<https://doi.org/10.1080/1553118X.2021.1887875>
- Friess, E. (2018). “Filling to Capacity”: *An Exploratory Study of Project Management Language in Agile Scrum Teams* (Vol. 65, Issue 2).
- Friess, E., & Lam, C. (2021). “Dude, that Sucks”: Examining Scrum’s Influence on Empathy in Student Teams. *Technical Communication Quarterly*, 30(2), 189–203.
<https://doi.org/10.1080/10572252.2020.1803413>

- Huang, C., & Kusiak, A. (1996). Overview of Kanban systems. *International Journal of Computer Integrated Manufacturing*, 9(3), 169-189.
<https://doi.org/10.1080/095119296131643>
- Hughey, D. (2009). The Traditional Waterfall Approach. Retrieved from Comparing Traditional Systems Analysis and Design with Agile Methodologies:
<https://www.umsl.edu/~hugheyd/is6840/Waterfall.html>
- Innosight. (2019, January 30). *Corporate longevity update: Creative destruction rides high*.
<https://www.innosight.com/insight/corporate-longevity-update-creative-destruction-rides-high/>
- KPMG International (2016). *KPMG's Global Transformation Study 2016: Succeeding in disruptive times*.
- KPMG International (2019). *Agile or irrelevant: Redefining resilience*.
- Linders, B. (2015, January 12). *Lean and Agile leadership with the scaled Agile framework (Safe)*. InfoQ. <https://www.infoq.com/news/2015/01/lean-Agile-leadership-safe/>
- Neill, M. S. (2018). *Change Management Communication: Barriers, Strategies & Messaging*.
<https://www.researchgate.net/publication/327057178>
- Opt, S., & Sims, C. D. L. (2015). Scrum: Enhancing Student Team Organization and Collaboration. *Communication Teacher*, 29(1), 55–62.
<https://doi.org/10.1080/17404622.2014.939675>
- Pope-Ruark, R. (2015). Introducing Agile Project Management Strategies in Technical and Professional Communication Courses. *Journal of Business and Technical Communication*, 29(1), 112–133. <https://doi.org/10.1177/1050651914548456>

- Project Manager/ScrumMaster, F. (2022, June 29). A Manager's Life With Agile. (J. Iwinski, Interviewer)
- Ragas, M., & Ragas, T. (2021). Understanding Agile for Strategic Communicators: Foundations, Implementations, and Implications. *International Journal of Strategic Communication*, 15(2), 80–92. <https://doi.org/10.1080/1553118X.2021.1898147>
- Rigby, D., Sutherland, J., & Noble, A. (2018, May/June). Agile at Scale. *Harvard Business Review*, 93(3), 89-96.
- Rovce, W. W. (1970). *MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS*. <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>
- Scrum.org. (2022). *What is Scrum?* <https://www.Scrum.org/resources/what-is-Scrum>
- Seiffert-Brockmann, J., Einwiller, S., Ninova-Solovykh, N., & Wolfgruber, D. (2021). Agile Content Management: Strategic Communication in Corporate Newsrooms. *International Journal of Strategic Communication*, 15(2), 126–143. <https://doi.org/10.1080/1553118X.2021.1910270>
- Takeuchi, H., & Nonaka, J. (1986). The new new product development game: Stop running the relay race and take up rugby. *Harvard Business Review*, 64(1), 137–146. <https://hbr.org/1986/01/the-new-new-product-development-game>
- van Ruler, B. (2015). Agile public relations planning: The Reflective Communication Scrum. *Public Relations Review*, 41(2), 187–194. <https://doi.org/10.1016/j.pubrev.2014.11.008>

ACADEMIC VITA

Jacob A. Iwinski

Pittsburgh, PA 15642

EDUCATION

The Pennsylvania State University | Schreyer Honors College
College of Liberal Arts | The Paterno Fellows Honors Liberal Arts
Majors: B.S. in Economics & B.S. in Communication Arts & Sciences
Minors: Business & the Liberal Arts, Business Certificate

University Park, PA
Class of 2022

PROFESSIONAL EXPERIENCE

RE Farm Café & Windswept Farm
Table Service & Special Events Server

University Park, PA
Aug 2021 – Present

- Analyzed restaurant operations data and presented findings to management
- Recognized by customers for outstanding quality of service which resulted in personal requests 6 times
- "Everything was perfect. Our server, Jacob, was a treasure" - Email to management

The Center for Performing Arts at Penn State
Official Photographer / Senior Photography Intern

University Park, PA
Feb 2017 – Present

- Led 6 photography workshops for university staff and students
- Operated in Beaver Stadium as an on-field photographer for the Penn State Blue Band
- Noted by management for excellence in performance/event photography

Office of Residence Life at Penn State
Resident Assistant

University Park, PA
Aug 2019 – May 2021

- Honored by the HRHH for outstanding contributions (hosted events) to the PSU Residence Hall Community
- Responsible for the safety, conduct, and community of 120 honors undergraduate residents over 3 years
- Won an award in a national contest for a bulletin board design out of 300+ of entries

Smeal College of Business
Teaching Assistant: Management 461 – International Management

University Park, PA
Aug 2019 – May 2021

- Prepared 30 presentations, lesson plans, assignments, and carried out data entry for finalization of grades
- Addressed student questions and assisting with tutoring/learning outside of class
- Facilitated in-class case studies, presentations, and attendance to reduce professor workload

Blue Water Growth LLC
Client Research Summer Intern

Pittsburgh, PA
Jun 2019 – Aug 2020

- Assessed 14 acquisition prospects on an environmental remediation project for clients of BWG LLC
- Identified 43 potential buyers for one of the world's largest private signed guitar collections (400+)
- Assembled report on 15 potential distributors in a medical device private capital raise

CAMPUS INVOLVEMENT

The Penn State Monty Python Society
Founder | President | Officer of Officers

University Park, PA
Jan 2019 – Dec 2021

- Founded a student organization to encourage comedic performance art like "Squirrel Fishing"
- Awarded most Innovative Program in 2020 out of 2000+ student clubs
- Recruited and managed a team of 6 officers that increased membership from 11 to 76 members
- Led and performed events to public audiences 24 times

AWARDS, SKILLS, AND INTERESTS

- **Awards:** Eagle Scout – Scouting BSA | Dean's List - College of Liberal Arts: 2017 - 2019 | Weiss Family Breakthrough Scholarship | The Pechter Family Scholarship | Shibley Raymond Memorial Scholarship
- **Technical Skills:** Microsoft Office: Word, PowerPoint, Excel | Industry Software: STATA, Python, Tableau | Adobe Suite: Photoshop, Lightroom, Bridge, Illustrator
- **Passions:** Photography, Rollerblading, Motorcycles, Manual Espresso, Mechanical Typewriters