Presented to the Interdisciplinary Studies Program:

UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

Best Practices for Implementing a Hybrid Project Management Methodology

CAPSTONE REPORT

Olivia L. Hillaire
Senior Business Systems Analyst
Intel Corporation

University of Oregon
Applied Information
Management
Program

Fall 2018

Continuing and Professional
Education
1277 University of Oregon
Eugene, OR 97403-1277
(800) 824-2714

Approved by

Dr. Kara McFall
Director, AIM Program

Best Practices for Implementing a Hybrid Project Management Methodology

Olivia L. Hillaire

Intel Corporation

Abstract

Best practices in implementing a hybrid project management methodology that will align stakeholders with differing project approaches were gathered from scholarly literature from 2005 to 2018. Project managers, directors, leaders, and customers will find this study useful because the sources identified provide key insights and background on Waterfall and Agile frameworks as well as the challenges and best practices in implementing a hybrid methodology.

Keywords: project management, hybrid, waterfall project management, agile project management, hybrid project management, blended project management, traditional project management

Table of Contents

Introduction to the Annotated Bibliography

Problem Statement

The Project Management Organization (PMO) in the Corporate Services (CS) division at Intel Corporation is struggling to blend Waterfall and Agile methodologies while implementing large-scale software solutions. While Intel's Information Technology (IT) department has recently recognized the benefits of Agile ideology for the development and implementation of software solutions and subsequently transformed into a fully Agile organization, not all departments and projects are a good fit for Agile. The work of many Intel departments aligns with and is managed by strictly Waterfall techniques; one example is construction projects. The PMO supports customers from both business and operational departments and requires flexibility to employ for some customers – those whose projects do not involve software solutions – the traditional Waterfall project life cycle's (PLC's) focus on scope, schedule, and budget and update customers on progress towards project milestones.

The disparity in operational styles between the IT teams who have embraced Agile methodologies and other customers whose projects involve software solutions and who prefer the traditional Waterfall approach has created a significant challenge for Intel's PMO that is charged with managing the projects and contributing to the solutions being developed by the Agile IT team. Particularly challenging is keeping stakeholders aligned and guiding customers, with their desires to define the schedule, cost, and requirements of the project up-front, through the process of ever-changing Agile requirements.

The Waterfall method was originally proposed by Winston W. Royce in 1970 as a potential software engineering practice (Bassil, 2012). Larman and Basili (2003) note that the methodology described in Royce's 1970 article, "Managing the Development of Large Software

Systems," would eventually "become known as the waterfall model, expressed within the constraints of government contracting at that time" (Larman & Basili, 2003, p. 48). Royce's paper became the standard of strict modern Waterfall sequencing (Larman & Basili, 2003).

Waterfall project management in the context of software development projects is defined as "a sequential software development process in which progress is regarded as flowing increasingly downwards (similar to a waterfall) through a list of phases that must be executed in order to successfully build a computer software" (Bassil, 2012, p. 2). The Waterfall model is characterized by distinct and easily recognizable project life cycle phases and well-defined planning and control methods (Hass, 2007). "Tasks are completed one after another in an orderly sequence, requiring a significant part of the project to be planned up front" (Hass, 2007, p. 2).

The benefits of employing a Waterfall approach to project management include the ability to estimate all of the work required to complete the project, assess costs of and resources necessary for that work, and finally control the costs and work throughout the project life-cycle (Rodrigues, 1996). The integration of subject matter experts (SMEs) from the organization into the project team and the clearly defined work structure allow for continuous control and planning activities as the implementation proceeds (Rodrigues, 1996). The Waterfall method is "one of the simplest software process models for application development, but it is also known for its ease of implementation in the field of software development" (McCormick, 2012, p. 6). For example, the model is designed in a linear fashion wherein you move to the next phase only when the current phase is complete, which promotes the definition of a clear work breakdown structure and the end goal early in the PLC and facilitates the measurement of progress (McCormick, 2012).

Challenges with a Waterfall approach include the methodology's inflexibility towards change, the amount of rework on a typical Waterfall project, and unpredictable software quality due to the delay of testing until late in the project (Petersen, Wohlin, & Baca, 2009). While there are numerous tools available for Waterfall project management, they do not supersede the need for adequate project and portfolio models and clear, consistent leadership driving the efforts (Sommer, Dukovska-Popovska, & Steger-Jensen, 2014). Sommer et al. (2014) identified a lack of systematic portfolio management in organizations that use Waterfall methodologies for project management because the projects of the loudest advocates were prioritized rather than the executive board prioritizing the most critical projects based upon documented criteria. Resource management can also be inconsistent on Waterfall projects because the best people are assigned the vast majority of the tasks (Sommer et al., 2014).

The field of Agile project management arose as a response to the increasing complexity within Waterfall projects that also experience uncertainty around outcomes and changing goals (Ciric, Lalic, Gracanin, Palcic, & Zivlak, 2018). In the context of this research project, Agile project management (APM) is defined as "a highly iterative and incremental process, where developers and project stakeholders actively work together to understand the domain, identify what needs to be built, and prioritize functionality" (Hass, 2007, p. 3). This approach allows IT developers and stakeholders to continuously evaluate functional priorities and make adjustments to requirements accordingly (Hass, 2007).

In 1986, Hirotaka Takeuchi and Ikujiro Nonaka published an article in Harvard Business Review (HBR); they described their study of "manufacturers that were releasing successful innovations far faster than competitors" and "identified a team-oriented approach that changed the design and development process for products such as copiers at Fuji-Xerox, automobile engines at Honda, and cameras at Canon" (Rigby, Sutherland, & Takeuchi, 2016a, para. 4). The authors found that these companies were not using traditional Waterfall project management methods “in which one group of functional specialists hand off its completed phase to the next functional stage” (Rigby et al., 2016a, para. 4). Instead, they were using techniques that would become the foundation for a specific Agile methodology called Scrum. Scrum is a rugby reference that describes how “a team tries to go the whole distance as a unit, passing the ball back and forth” (Rigby et al., 2016b, para. 4), much like the developers at Fuji-Xerox, Canon, and Honda were doing to release products faster than their competitors.

In 1993, Jeff Sutherland, who would later co-author The Manifesto for Agile Software Development and co-invent the Agile methodology called Scrum, was helping Easel Corporation, a software development firm, develop a new product to replace its legacy products, with a timeline of less than six months (Rigby et al., 2016a). Sutherland had a strong background in software development and project management methodologies, which allowed him to learn everything he could about maximizing organizational productivity, and “he found himself intrigued by several provocative ideas” related to what would later be formalized as Agile development processes (Rigby et al., 2016a, para. 5).

Sutherland and several prominent practitioners of software development convened in 2001 “to arrive at a consensus on how the software development industry could produce better results and overcome limitations of traditional software development process” (Ciric et al., 2018, p. 2). They created the Manifesto for Agile Software Development, which contained twelve principles and four values and expressed the basic ideology of Agile software development (Ciric et al., 2018). The practitioners’ goal was to “increase the quality, enhance flexibility and accelerate time to market” (Ciric et al,, 2018, p. 2). The Agile Manifesto creators sought to address limitations of traditional software processes, such as the lack of efficacy within traditional project management methodologies in effecting change during a project or taking advantage of new opportunities as they emerge during the project lifecycle (Ciric et al., 2018).

The four values of the Agile Manifesto are as follows:

1. "**Individuals and interactions** over processes and tools,
2. **Working software** over comprehensive documentation,
3. **Customer collaboration** over contract negotiation, and
4. **Responding to change** over following a plan" (Beck et al., 2001, para. 1).

Beck et al. (2001) note that the emphasis on the items highlighted on the left reflects the fact that they value them more than the items on the right.

Rigby, Sutherland, and Takeuchi (2016b) note the benefits of Agile methodologies include the following: (a) enables radical breakthroughs while retaining the benefits that accrue with operating as part of a parent organization, (b) delivers the most valuable innovations earliest in the project, (c) facilitates the rapid increase of team happiness, and (d) enables team members to develop general management skills. Agile methods also eliminate what Karlström and Runeson (2005) refer to as the "requirements cramming" problem, wherein customers request extra features during the development phase (p. 46). In stage-gate development, these change requests are often included as additional features in the project plan rather than trade-offs, thus bloating the scope (Karlström & Runeson, 2005). This practice did not allow much insight into the gradually increasing workload or uncertainty about the overall progress of the project due to scope creep (Karlström & Runeson, 2005).

Increased team happiness on Agile projects may result from the fact that, "unlike the traditional methodologies, agile methodologies deal with unpredictability by relying on people and their creativity rather than on processes" (Nerur, Mahapatra, & Mangalaraj, 2005, p. 75).

Team members are empowered to address challenges, thus leading to increased satisfaction (Nerur et al., 2005). Agile seeks to respond to change rather than viewing it as a constraint (Boehm & Turner, 2005). For example, “Product functionality might change in order of delivery, or the agile team might change its design specifics as properties emerge and the team incorporates customer feedback” (Boehm & Turner, 2005, p. 31). Changing requirements are therefore seen as an opportunity to meet the customer’s needs rather than an obstruction to project success (Boehm & Turner, 2005). In addition, working software serves as a better measure of progress and detection of low-value functionality occurs earlier and is thus prevented (Boehm & Turner, 2005).

While Agile methodologies have many proponents, there are also recognized challenges (Rigby et al., 2016b). Rigby et al. (2016b) note that “agile is not a panacea” (para. 14). Tools are critical to Agile project success, and Nerur et al. (2005) note that an organization’s existing technology may hinder attempts to migrate to Agile methodologies. For example, “companies that rely solely on mainframe technologies may find it difficult to assimilate agile methods compared to those that use OO [object-oriented] development techniques” (Nerur et al., 2005, p. 77). Agile requires an investment in tools that support the fast pace and facilitate the iterative nature of Agile development techniques such as versioning management and refactoring, as well as the training required so that project members use the tools correctly (Nerur et al., 2005).

Boehm and Turner (2005) list many barriers, both real and perceived, to implementing Agile processes. Variability in development approaches and deliverables is a concern; when “both agile and traditional teams are developing software for the same product, they can develop radically different artifacts that might not integrate easily” (Boehm & Turner, 2005, p. 2). Similarly, Boehm and Turner (2005) note that using Agile processes for legacy systems raises multiple issues, including the difficulty in disassembling a system in a way that accommodates incrementally built replacements and the difficulty refactoring the embedded business processes that are now institutionalized in the culture (Boehm & Turner, 2005).

Dingsøyr et al. (2018) describe an issue on Agile projects where Business Analysts had increased workloads as they focused on upcoming releases. Their analysis work on business processes and product features on current releases was put aside as priorities shifted, and they had to change focus and prepare new analyses for the newer, recently committed work (Dingsøyr et al., 2018). Dingsøyr et al. (2018) also note that "when scaling agile development our experience is that you need additional roles and arenas to ensure efficient coordination between the teams" (p. 4). Additional roles include a dedicated technical architect as well as a functional architect and a mix of senior and junior developers (Dingsøyr et al., 2018).

Hybrid project management arose out of the desire to realize the benefits of Agile methodologies while still maintaining some of the structure of Waterfall methodologies (Špundak, 2014). Hybrid project management is defined as "the borrowing, mixing, and blending [of] processes from both agile and plan-driven methods to tailor project specific methodologies" (Schmitz, Mahapatra, & Nerur, 2018, p. 1). Benefits of a hybrid methodology include flexibility for the project team to customize a project-specific combination of methodologies (Schmitz et al., 2018). For example, on a project with low overall variability where the customer has put an extremely high value on the product, a project team may choose Agile development methods and frequent user engagement and testing, combining this iterative approach in development/testing with Waterfall PLC management and a phased progression (Nerur et al., 2005). Overall variability is determined through the combined variability of three factors: tasks, capability of the assigned human resources, and the technology being used by team members to develop the project's deliverables (Nerur et al., 2005).

The same characteristics that are evident in Waterfall or Agile project management methodologies may be found in hybrid project management, but are applied to a project on a case-by-case basis (Špundak, 2014). However, the blending of Waterfall and Agile methods can prove challenging (Špundak, 2014). Špundak (2014) described one example of a software project performed by a small IT firm that was used to Agile processes for a large bureaucratic company that was used to employing Waterfall methodologies. Špundak (2014) observed that while the bureaucratic customer organization required the formal documentation commonly found in Waterfall project management, the Agile IT firm’s reliance on tacit knowledge did not satisfy this requirement (Špundak, 2014). The Agile IT firm needed to document their knowledge in a manner sufficient to satisfy the bureaucratic company’s existing business processes (Špundak, 2014). Other areas of dissimilarity between Waterfall and Agile that necessitate conscious choices occur in the categories of expected changes to requirements, the role of users on the project, whether to scale a project to a small size to accommodate Agile or to go larger where Waterfall works better, the type and amount of organizational support, and the team members selected along with their roles within the project (Špundak, 2014).

Hass (2007) notes that the principles of either traditional or Agile project management “can be implemented into most team-structured environments” (p. 5). Hass (2007) recommends that organizations that embrace Waterfall methodologies do not reject all Agile principles, noting that often some Agile principles can be successfully adopted. For example, Hass (2007) believes a project should involve the users to the extent that they are available, even if only for testing or prioritizing features.

This research study explores best practices in employing hybrid project management. Literature sources that are included in the annotated bibliography provide information on the background and characteristics of Waterfall and Agile methodologies to provide a context for the topic of best practices in the implementation of a hybrid project management approach. Sources on the best practices of a hybrid project management approach include information on best practices including tools and processes.

Purpose Statement

The purpose of this qualitative study is to identify and describe the best practices for blending Waterfall and Agile project management methodologies to aid in the alignment of stakeholders with differing project approaches. The design of the study is a literature review and the method of inquiry is the collection, sorting, review, annotation and analysis of selected research articles. The primary audience is the members of the Projects and Solutions Team (PST) within the Corporate Services (CS) PMO of Intel Corporation, although other organizations and departments who are considering implementing a hybrid approach or are in the midst of a hybrid implementation will benefit. The goal of this annotated bibliography is to provide this audience with resources that provide best practices in blending Waterfall and Agile project management methodologies.

This study fulfills multiple goals. First, sources in the study provide an historical context for Waterfall and Agile project management methodologies. Second, the sources describe the best methods for successful implementation of a hybrid project management approach and the expected and unexpected outcomes of others who have concluded their own implementations. Next, the sources provide baseline descriptive data about specific actions that demonstrate higher or lower levels of effectiveness in achieving successful project outcomes when utilizing a hybrid approach. Finally, the findings serve as an initial foundation upon which to build a playbook to apply best practices for the implementation of a hybrid project management methodology in an existing organization.

Research Question

Main question. What are the best practices in implementing a hybrid project management methodology that will align stakeholders with differing project approaches?

Sub-questions. What tools from Waterfall and Agile can be successfully used in hybrid projects? What obstacles need to be overcome by both project managers and customers when implementing a hybrid project management approach?

Audience

The intended audience for this study is the management team within the Corporate Services (CS) division of Intel Corporation, as well as the Project Managers in the Projects and Solutions Team (PST) within CS. Intel's CS division is structured with one manager who has twelve directors heading functional areas such as Construction, Operations, and Health & Safety. PST has four sub-teams: Project Managers, Business Analysts, Communications, and Data Analytics. There are eleven project managers with varying levels of experience.

Each functional area director will often be the ultimate decision-maker during a project, especially on issues related to budget. These managers need this study to understand both the benefits and challenges facing our PMO while we determine the appropriate blend of Waterfall and Agile methodologies to enable the successful implementation of the solutions their functional areas are requesting. The functional area managers are our advocates during future projects and are imperative to creating a culture where the ambiguity inherent in Agile methodologies is accepted.

The project managers hold most of the responsibility for adhering to scope, schedule, and budget throughout the PLC. The best practices and tools identified in this study will enable them to navigate their responsibilities more effectively as they transition from employing only Waterfall methodologies to a hybrid approach.

Outside of Intel, the audience for the study consists of Project Managers, Directors and other leaders who are using or considering using hybrid project management approaches, and project customers who want to understand their roles on hybrid projects. There are many in the field of project management facing similar struggles in forming hybrid methodologies to apply to their projects. The literature in this study will help the audience understand the universality of the challenges they are experiencing and provide best practices for meeting those challenges.

Search Report

Search strategy. I conducted research through the University of Oregon (UO) Research Library for Computer Science and Information Systems. I collected references that focus on methods of Waterfall project management, Agile project management, and hybrid project management. Sources that offered information about any of these three categories helped to develop my understanding of the histories, benefits, challenges, and impacts of these methods. Sources that illustrate best practices of hybrid project management were critical to the success of my research.

Databases. I primarily used Google Scholar and the Institute for Electrical and Electronics Engineers (IEEE) Xplore Digital Library in searching for sources.

Search terms. My list of search terms included the following:

- History of Waterfall.
    - Waterfall project management,

  - Traditional project management,
  - Stage-gate project management,
  - History of Waterfall project management,
  - Benefits of Waterfall project management,
  - Challenges of Waterfall project management, and
  - Impacts of Waterfall project management.
- History of Agile.
  - Agile project management,
  - History of Agile project management,
  - Benefits of Agile project management,
  - Challenges of Agile project management, and
  - Impacts of Agile project management.
- Best practices in implementing a hybrid approach.
  - Agile Waterfall project management,
  - Traditional project management agile development,
  - Waterfall hybrid methodology,
  - Agile hybrid methodology,
  - Hybrid project management methodology, and
  - Successful / effective project management.

Documentation strategy. I stored and organized articles and notes using Microsoft OneNote software. I utilized nested Section Groups to group by category and the pages within each section to track the necessary American Psychological Association (APA) citation information. I saved a portable document format (PDF) copy of each article in a folder in

Dropbox. I then imported each source into OneNote and made notes in the margins. I added the relevant APA information to OneNote to form the Bibliography and Reference lists, including author name(s), date of publication, title, source, digital object identifier (doi), Uniform Resource Locator (URL), and abstract. I used a table within OneNote to track all of the information from the sources and to note how I used the sources in the study.

Reference evaluation criteria. I evaluated sources with the criteria recommended by the University of Florida's Center for Public Issues Education (2014). This resource stresses the importance of the skill involved in evaluating information and deciding which sources to accept (University of Florida's Center for Public Issues Education, 2014).

Timeliness. For sources related to the characteristics of Agile and hybrid project management, I began my searches by limiting the year of publication to 2018. I found limited sources initially but got meaningful results as I expanded my search one year at a time. The final publication dates for the body of work for my research range from 2005 to 2018. I found sources within this timeframe that provide key insights on both the Agile framework as well as the challenges and best practices in implementing a hybrid methodology.

I tried this same method with Waterfall project management but I struggled to find sources until I expanded the search criteria to include older dates. Petersen et al. (2009) found that "the [Waterfall] model seems to be of very little interest for researchers to focus on as it seems to be old-fashioned. Instead, recent studies have much more focus on agile and incremental development" (p. 1). In addition, the Waterfall methodology was first introduced in 1970 (Bassil, 2012), so finding current sources was less critical than for the more recent topics of agile and hybrid project management. Sources for Waterfall project management cover a range from 1996 to 2017. I had to expand my search to get a sufficient amount of sources for a meaningful review.

Relevancy. I determined the relevancy of each source for this study by requiring that each article presented information aligned with the purpose of this study and was appropriate for the intended audience. I reviewed the reference lists for each source to confirm that the author(s) consulted scholarly works related to the purpose of their publication.

Quality. I considered writing standards that demonstrate accuracy in grammar, spelling, and punctuation and concentrated my research on peer-reviewed sources.

Authority. I evaluated sources for their scholarly merits and rejected authors with no apparent authority in the subject area. I also determined authority by noting citations for the author in other works and number of published works on the given subject, as well as the author's years of experience in the industry. Any documentation with an anonymous author was attributed to the authority of the organization presenting the material.

(Lack of) Bias. I rejected content that I assessed as propaganda. I reviewed the bibliography of each source presenting facts or opinions to confirm whether those facts or opinions were supported in trusted sources. I reviewed the literature and rejected those with emotional language, broad generalizations, oversimplifications, or one-sided treatments of a subject.

Annotated Bibliography

This annotated bibliography provides references that address the best practices for implementing a customized hybrid project management methodology. The references are categorized as relating to one of three subjects: (a) characteristics of Waterfall project management, (b) characteristics of Agile project management, and (c) best practices in implementing a hybrid approach. The abstracts presented are the published abstracts. The summaries were written by the author of this annotated bibliography, but contain content, ideas, and findings pulled from each source.

Characteristics of Waterfall Projects

Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *Lebanese Association for Computational Sciences (LACSC), International Journal of Engineering & Technology (iJET), 2*(5), 742-749. Retrieved from https://arxiv.org/abs/1205.6904

Abstract: Software development life cycle or SDLC for short is a methodology for designing, building, and maintaining information and industrial systems. So far, there exist many SDLC models, one of which is the Waterfall model which comprises five phases to be completed sequentially in order to develop a software solution. However, SDLC of software systems has always encountered problems and limitations that resulted in significant budget overruns, late or suspended deliveries, and dissatisfied clients. The major reason for these deficiencies is that project directors are not wisely assigning the required number of workers and resources on the various activities of the SDLC. Consequently, some SDLC phases with insufficient resources may be delayed; while, others with excess resources may be idled, leading to a bottleneck between the arrival and delivery of projects and to a failure in delivering an operational product on time and within budget. This paper proposes a simulation model for the Waterfall development process using the Simphony.NET simulation tool whose role is to assist project managers in determining how to achieve the maximum productivity with the minimum number of expenses, workers, and hours. It helps maximizing the utilization of development processes by keeping all employees and resources busy all the time to keep pace with the arrival of projects and to decrease waste and idle time. As future work, other SDLC models such as spiral and incremental are to be simulated, giving project executives the choice to use a diversity of software development methodologies.

Summary: This source is relevant for this study because the author describes the Waterfall method and discusses its key challenges. In his article, Bassil explains that the Software Development Life Cycle (SDLC) has been studied by many researchers proposing various models which each have their own strengths and weaknesses. Bassil states that all the models share some basic properties, wherein they each have sequential phases that lead to results and deliver a final product. Bassil notes the Waterfall model is one of the earliest SDLC models, first introduced in 1970 by Royce, which has been adopted by many software development firms to plan, build, and maintain their products. He defines the Waterfall model as “a sequential software development process in which progress is regarded as flowing increasingly downwards (similar to a waterfall) through a list of phases that must be executed in order to successfully build a computer software” (p. 2).

Bassil explains each of the five phases of the Waterfall model as follows:

- Analysis phase provides a complete and comprehensive description of the behavior of the software to be developed. Business/systems analysts define all requirements (functional and non-functional). Functional requirements are defined by how the users will interact with the software. Non-functional requirements define constraints for the design/operation of the software and include reliability, scalability, testability, maintainability, performance, and quality standards.

- Design phase completes the planning and problem solving necessary to implement a software solution. Software developers define each component of the plan for a solution, such as software architecture design, database conceptual schema and logical diagram design, concept design, graphical user interface design, and data structure definition.
- Implementation phase refers to the realization of all of the requirements and design specs into a usable tool. This phase is where the real code is written and where the database and text files are created (i.e. the process of converting requirements and design into a production environment).
- Testing phase verifies and validates that the software solution meets the original requirements and specs and that it meets its intended purpose. Debugging is performed wherein system glitches are found, corrected, and refined accordingly.
- Maintenance phase includes modifying a software solution after deployment to correct for any issues that arise or to implement enhancements, which may include adapting software to its environment, accommodating new user requirements, and increasing software reliability.

Larman, C. & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer 36*(6), 47-5*6.* https://doi.org/10.1109/MC.2003.1204375

Abstract: Although many view iterative and incremental development as a modern practice, its application dates as far back as the mid-1950s. Prominent software-engineering thought leaders from each succeeding decade supported IID practices, and many large projects used them successfully. These practices may have differed in their details, but all had a common theme-to avoid a single-pass sequential, document-driven, gated-step approach.

Summary: This article is relevant for this study because it provides a strong history of the Waterfall methodology, explaining its origins along with those of iterative and incremental development methodologies. While the article was published in 2003, as previously established, the Waterfall methodology has not changed substantially in recent years. The authors describe Winston Royce's role in the creation of the Waterfall method during the 1970s. The authors explain that Royce's model was meant to be iterative, wherein he recommended that the released version of the software be the second version of critical design. The authors state that the hints of iterative development, feedback, and adaptation are further suggested by Royce's proposed 10-month pilot model within a 30-month development project. The authors contend that the strict sequence method described by Royce was incorrectly adopted as single-pass, which is the form Waterfall takes to this day in most project management efforts. The authors conclude that while Waterfall was never intended to be a true iterative and incremental development (IID) approach, the original elements that included iterative feedback have been lost.

This interpretation of Royce's Waterfall method can be useful in demonstrating how smoothly and effectively the Waterfall method can be adapted to an iterative Agile approach. The authors' findings provide strong evidence to build upon when drafting a solution to the problem statement described in this study.

The authors provide the history of several iterative and incremental development projects and approaches categorized chronologically from pre-1970 through 2003. The authors note that during this timeframe IID concepts remained a recommended practice and were often associated with the success of large projects. The authors recommended an increase in education and promotion of IID methods to help convince organizations, consultants, and standards bodies that were still promoting a single-pass sequential life cycle as the ideal of the benefits of Agile methods. This advice is still useful today, as there are still many proponents of the single-pass sequential life cycle.

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. *International Conference on Product-Focused Software Process Improvement--PROFES 2009,* 386-400. https://doi.org/10.1007/978-3-642-02152-7_29

Abstract: Waterfall development is still a widely used way of working in software development companies. Many problems have been reported related to the model. Commonly accepted problems are for example to cope with change and that defects all too often are detected too late in the software development process. However, many of the problems mentioned in literature are based on beliefs and experiences, and not on empirical evidence. To address this research gap, we compare the problems in literature with the results of a case study at Ericsson AB in Sweden, investigating issues in the waterfall model. The case study aims at validating or contradicting the beliefs of what the problems are in waterfall development through empirical research.

Summary: This source is relevant for this study because the authors conducted case study research on the usefulness of the Waterfall model to aid in decision-making of when to use the model. The authors' case study specifically focuses on identifying issues in the Waterfall methodology, observed at a real company, and compares these findings with problems related to Waterfall that appear in literature. The authors studied the use of Waterfall at a Swedish Ericsson AB development site that had been using Waterfall for several years. The authors state that their study contributes to research on the Waterfall model in the following ways: (a) Illustration of the model in practice at a large-scale industrial software development organization, (b) Identification and prioritization of issues related to the Waterfall model, and (c) Comparison of the case study results with current literature.

The authors note that Ericsson AB used a stage-gate Waterfall method, which means that there is an official quality check between each phase of the Waterfall model. The authors describe 13 issues they found, prioritized into four tiers, labeled A, B, C, and D, from highest criticality to lowest. The authors then compared these issues to those they had found in the literature, and report that they did not find any issues in the literature that did not also show up in their case study.

The authors identified four issues not mentioned in any literature, two from each of the C and D tiers respectively, which were the lowest criticality tiers. The new issues are described as follows:

- Confusion of who implements which version of the requirements,
- High effort required for maintenance of the resulting code,
- Lack of broadly skilled employees, and

- Problems identifying the root cause of a defect due to communication barriers.

Overall, the most critical issues they identified relate to requirements (analysis phase) and verification (testing phase). Specific issues with requirements included the creation of too much documentation during requirements gathering that was never used in later stages of the project, as well as the fact that the impact of new requirements was not understand as they related to other parts of the system. Specific issues with verification included the need to limit testing time at the end of the project as a result of other phases running past schedule and the fact that defects that are discovered late are more difficult and more expensive to fix. The authors conclude that their findings describe the issues more thoroughly than the literature they researched and noted their identification of four new issues related to the use of the Waterfall methodology. The authors' thorough exploration of common issues with the Waterfall method will be helpful when looking at solutions, as their findings can help to illuminate pitfalls with the Waterfall methodology so that mitigations can be identified.

Rodrigues, A. & Bowers, J. (1996). System dynamics in project management: A comparative analysis with traditional methods. *System Dynamics Review 12*(2), 121-139. https://doi.org.10.1080/08874417.2009.11646044

Abstract: Recent dramatic project failures have highlighted weaknesses in the traditional approaches to project management and in particular their failure to cope with strategic issues. System dynamics models provide a useful tool for a more systematic management of these strategic issues. There have been a number of applications of system dynamics in project management; this experience permits a tentative comparison with the more traditional approaches and to examine the particular benefits of system dynamics. The conflicts of opinion between their supporters stress the different perspectives underlying the two approaches. The comparison of the approaches is focused on the "view" of the project management process. Although, ultimately, they both assume a system perspective, identifying a cycle of planning, implementation and control, the level of detail in which they consider the project system is different. Traditional models support the project manager in the detailed operational problems within the process, while system dynamics models provide more strategic insights and understanding about the effectiveness of different managerial policies. The two approaches provide complementary support to project management; this suggests it could be of major value to integrate the best of both worlds.

Summary: This source is relevant for this study because the authors acknowledge the importance of project management but delve into the deficiencies of traditional techniques in the hopes of finding an alternative. The authors contend that “traditional techniques can encourage a narrow, operational view of the project, concentrating on the detailed planning” (p. 121). The authors note that several studies, including those performed by Davidson and Huot (1991) and Morris and Hough (1987), have identified the need for a more strategic approach to project management. The authors list the primary tools and techniques of traditional project management methods, as well as their roles in project management. These tools include the work breakdown structure (WBS), which provides the basic definition of the project work; the responsibility matrix, which assigns project team members to the work listed in the WBS; Gantt chart, which provides a visual representation of the project schedule; project network techniques that illustrate scheduling impacts that activities have on each other and provide the means to determine critical activities and float times; cost scheduling, which identifies the necessary capital investment; and a project control process, which will detect project overruns in schedule or budget.

The authors state that traditional project management is based on the classic control cycle, and describe a dynamic control process that takes place within a well-ordered project system that interacts with external environmental elements. The authors assert that well-defined, linear stages of traditional methods, inherited from the construction industry, may not be optimal for design and development projects that require a more iterative nature.

The authors suggest that projects are increasingly complex and play an ever more important role in an organization. They contend that a focus on strategic, rather than short-term, planning is crucial to project success and that decisions should have some discernible, pervasive logical pattern because of the number of activities on which strategic decisions have an effect. The authors suggest that “a project’s strategy should define its position relative to its environment recognising [sic] the critical constraints; the strategy should ensure that the project contributes to the organisation’s [sic] long term objectives” (p. 123).

The authors state that most traditional techniques focus on a tactical-level approach to project management. They contend that this may work well for operational issues which have clearly defined objectives and constraints, but suggest this is not sufficient for every level of the project and that strategic analysis will require more flexible tools in order to model more complex, less quantifiable issues.

The authors explore the nature of project failure as internal to the project rather than as a result of external factors, which are usually blamed. The authors contend that the management methodology used, coupled with ineffective organizational processes, is likely to be blamed when projects fail. The authors suggest that good project management should be able to handle negative external influences and lead to successful completion in spite of them.

The authors contend that a strategic perspective will help project managers cope with the human failures (so called *soft factors*) common to traditional techniques but that personal judgement is not sufficient due to the unpredictable and unstable nature of projects. The authors suggest that a better understanding of the strategic issues within a project requires a formal, systemic analysis tool.

The work within this study will be useful because it demonstrates that a structured project management model can be flexibly applied to each unique project. The elements of a structured project management model can be evaluated for their effectiveness within any given project. The solution developed from this study should emphasize a process as well as the cultural elements necessary to transfer lessons from past mistakes into future practice.

Characteristics of Agile Projects

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M.,…Thomas, D. (2001). The agile manifesto. Retrieved from https://moodle2016-17.ua.es/moodle/pluginfile.php/80324/mod_resource/content/2/agile-manifesto.pdf

Abstract: Abstract written by the author of this annotated bibliography in the absence of a published abstract. In February 2001, seventeen people met at a lodge in Utah to try to find common ground. The people represented Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, and Pragmatic Programming. What they created was the Agile Manifesto. The Manifesto provides some specific ideas but a deeper theme drives many of the Agile Alliance members. The Alliance was comprised of people who held up a set of values based on trust and respect for each other. These values promoted collaborative organizational models in which they would want to work.

Summary: This source is relevant for this study because it is a foundational text on the topic of Agile methodology. The authors include the four values of the Agile Manifesto as follows:

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

The authors emphasize the content on the left of each value because "While there is value in the items on the right, we value the items on the left more (para. 3).

The authors also provide a more detailed list to guide those who wish to adopt Agile; the twelve principles of Agile are:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.
9. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The authors explain that Agile embraces the difficult trade-off decisions that many organizations avoid through power struggles and by imposing irrational demands. The authors suggest that the Agile movement is not anti-methodology; in fact, it seeks to restore credibility to the word methodology. The authors state that they embrace both modeling and documentation, but not when the outputs are stored in dusty basements and not used. The authors assert that they embrace planning, but recognize the limits of planning in ambiguous, turbulent environments. The authors note that ultimately they seek to develop quality products. The authors conclude that the Agile Alliance seeks to help others think about software development, methodologies, and organizations in new and more agile ways.

Boehm, B. & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software 22*(5), 30-39. https://doi.org/10.1109/MS.2005.129

Abstract: Discussions with traditional developers and managers concerning agile software development practices nearly always contain two somewhat contradictory ideas. They find that on small, stand-alone projects, agile practices are less burdensome and more in tune with the software industry's increasing needs for rapid development and coping with continuous change. Managers face several barriers, real and perceived, when they try to bring agile approaches into traditional organizations. They categorized the barriers either as problems only in terms of scope or scale, or as significant general issues needing resolution. From these two categories, we've identified three areas - development process conflicts, business process conflicts, and people conflicts - that we believe are the critical challenges to software managers of large organizations in bringing agile approaches to bear in their projects.

Summary: This source is relevant for this study because the authors explore the challenges inherent during an implementation of Agile methodology. Specifically, the authors focus on the barriers to scaling up and integrating Agile when an organization already has traditional, top-down systems development practices in place. The authors explain three areas as the most critically challenging to those managing software projects in large organizations who are implementing Agile techniques: development process conflicts, business process conflicts, and people conflicts.

The first area the authors explore is the development process conflicts found while merging lightweight Agile processes with standard industrial processes. The authors suggest an organization runs the risk of either killing agility or undermining the processes they spent years refining. In order to be successful, the authors assert that an organization must consider some key components: variability in subsystems and teams, different life cycles, legacy systems, and requirements. The authors say that managing variability in subsystems and teams is difficult and larger organizations need to pay attention to how they synchronize teams. For example, some organizations were successful using medium-sized teams, structured for each organization into a team of teams, and ensuring each team was assigned a competent team leader possessing an appropriate mix of technical, people, and Agile skills.

The authors state that the disparity of working with different life cycles can also be difficult since Waterfall focuses on a longer period of time and Agile delivers functionality more frequently. The authors suggest that an organization can adjust the Agile processes to help blend the methodologies, such as producing the documentation that traditional methods require. The authors also explain that an organization must consider replacing or enhancing legacy systems developed using Waterfall methods because they are not generally easy to refactor to accommodate Agile replacements or enhancements that build capability incrementally. The authors contend that an organization may need to strengthen their Agile requirements approach to include additional information beyond purely functional features, such as those dealing with reliability or security. The authors provide specific, actionable suggestions for mitigating the risks from developmental process conflicts, such as advising organizations to conduct significant analysis up front of all current state and proposed processes and to develop architecture in a way that supports the compartmentalization necessary for Agile teams.

Next the authors explore the business process conflicts that arise when implementing Agile in an organization that has previously embraced Waterfall methodologies because these organizations often require accurate predictions of tasks and do not encourage experimentation or evolution. The authors suggest that these approaches are difficult to integrate with iterative processes that rely on evolution. The authors describe key areas for concern: human resources, progress measurement, and process standard ratings. The authors assert that human resource issues include assessing the required skills and level of experience of team members and enabling a culture wherein rewards are based on team achievement rather than individual achievement. The authors note that traditional progress management techniques may be too slow to keep pace with Agile processes, which proceed rapidly. They state that using requirements burndown, which measures the quantity of requirements from the backlog that have been marked completed, or user story completions may be beneficial in this case and can be used as reasonable substitutes for traditional milestone reviews.

The authors also contend that safety-critical areas that require some form of certification and rigorous appraisals create a risk because Agile methods, with a focus on constantly adapting to improve performance, may not support the documentation required for certification. The authors provide specific, actionable suggestions for mitigating the risks from business process conflicts such as addressing HR issues early, researching how to modify legacy systems in order to enable Agile-compatible re-engineering or maintenance, and updating contracting practices so that they support Agile concepts.

Lastly, the authors contend that people conflicts are by far the most critical when implementing Agile in an organization used to Waterfall, and include management attitudes, logistical issues, handling successful pilots, and change management. The authors explain that traditional project management methods tend to see employees in specific roles, which does not align with Agile teams that rely on multitasking and task shifting. The authors note that logistical issues may arise because Agile development team members should be co-located with collaboration spaces so they can easily share information. The authors recommend handling successful Agile pilots with care to maintain team relationships, retain knowledge gained, and send the message that trying new things is encouraged. The authors describe the pain experienced by organizations when they try something new and employees display concerns that stem from feelings of inadequacy, jealousy, and defensiveness. They further note that these negative feelings can lead to negative behaviors that may undermine the new methods. The authors provide specific, actionable suggestions for mitigating the risks from people conflicts such as educating stakeholders on specific methodologies the team is using, emphasizing the value of the people selected for the projects, and rewarding the results they produce.

The authors conclude that the challenges in implementing Agile methodologies in an organization that previously utilized Waterfall methodologies are not insurmountable, but the shift does require diligence, patience, and hard work. They also conclude that more research is needed to find new approaches and discover ways to harmonize the methods. This article provides specific suggestions on how to deal with the most critical challenges facing organizations implementing Agile. These suggestions will be useful when framing a solution to the problem statement described in this study.

Ciric, D., Lalic, B., Gracanin, D., Palcic, I., & Zivlak, N. (2018). Agile project management in new product development and innovation processes: Challenges and benefits beyond software domain. *IEEE Xplore Digital Library.* https://doi.org/10.1109/TEMS-ISIE.2018.8478461

Abstract: Since the inception of Agile Manifesto and in the 15 years since agile project management (APM) has been in use, it has become a cutting-edge approach practiced, primarily although not exclusively, in the software industry. APM is slowly breaking through into other domains as well. Therefore, the objective of this study is to summarize and critically examine contributions of past research and existing knowledge concerning APM applicability beyond software industry and its co-existence with traditional project management (TPM) practice, and to identify the potential challenges and benefits of APM in innovation and new product development processes. Systematic literature review (SLR) protocol was developed to systematically locate, assess and aggregate the outcomes from all relevant studies in a transparent way. Gaps identified in this SLR will be used to suggest areas for further investigation and to provide a framework for appropriate position of future research activities.

Summary: This source is relevant for this study because it examines the research to date and existing knowledge on APM along with how it coexists with traditional project management methodologies. The authors include a list of benefits of including APM techniques: greater flexibility, reduced planning time, reduced cost and schedule, improved communications, and success at delivering what the customer expected. The authors also note the challenges often seen when blending Agile with Waterfall, including a lack of Agile culture in the organization, difficulty dedicating teams full-time to the project, management resistance, and mismatches between the requirements of Agile and the company's reward system. The authors contend that APM's adoption beyond the software industry is due to increasingly complex projects with higher levels of uncertainty, which lend themselves to Agile methods. The authors found that project managers are now more aware than they were in earlier times of the shortcomings of traditional project management methods in dealing with change and taking advantage of new opportunities. They assert that organizations must recognize project changes, such as shifting customer priorities, and be flexible in meeting them in order to gain or keep their competitive advantages. The authors suggest APM has emerged because it offers this necessary flexibility while remaining disciplined through highly-trained, skilled, and collaborative teams. The authors say that despite the benefits of Agile there is still a lack of understanding on how to identify when this methodology would be appropriate, how to apply it outside of software development, and how to effectively integrate it into current traditional project management practices.

The authors provide a background of the Agile movement and Agile Project Management, beginning with the introduction of the Agile Manifesto in 2001. The authors' research led them to the conclusion that other researchers were looking to find examples of APM being applied beyond the software industry. The researchers' specific interest was in the aforementioned benefits and challenges that arose when trying to apply APM to innovation and product development processes. The researchers recognized that integrating Agile practices depended on an organization’s environment and that success depended on the organizations having some characteristics similar to software companies. The researchers found that Agile methods improved the practices within an organization regarding how they transfer knowledge and understand the needs of their customers that can be applied to innovation.

The authors explain that an organization should understand the context of the organization and pull from a diversity of project management solutions as appropriate. The authors found that the Agile methodology is not a good fit for every organization or scenario, but is also not an all-or-nothing methodology; an organization can incorporate even small elements successfully.

The authors report that a limitation on their study was the lack of scholarly articles on implementing APM due to its recent emergence. The authors explain there is a lack of empirically grounded studies on the topic and those that exist are limited for reasons such as insufficient sample size, limited number of industries represented, and limited geographic regions, which makes it difficult to form generalizations. The authors report that the examples they found are useful but will only be relevant for similar organizations to those they investigated. The authors further report that their study was qualitative in nature and may have had a natural bias in the selection process or possible imprecision when they extracted data.

This source identifies the benefits and challenges of implementing Agile and explores specific organizational examples. This will be a useful contribution to support the identification of practical solutions to the problem statement described in this study.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM – Adaptive Complex Enterprises 48*(5), 72-78. https://doi.org/10.1145/1070000/1060712

Abstract. Software development methodologies are constantly evolving due to changing technologies and new demands from users. Today's dynamic business environment has given rise to emergent organizations that continuously adapt their structures, strategies, and policies to suit the new environment [12]. Such organizations need information systems that constantly evolve to meet their changing requirements---but the traditional, plan-driven software development methodologies lack the flexibility to dynamically adjust the development process.

Summary: This source is relevant for this study because it presents the benefits and challenges encountered by organizations going away from plan-driven or Waterfall methodologies and adopting various methodologies within the Agile ideology. In this journal article, the authors seek to understand the impacts of Agile on specific characteristics of existing Waterfall project management teams. The authors explain that the organizational structure in an environment where Agile will be successful is different than the organizational structure necessary for Waterfall's success. The authors state that organizational structures characterized by strong leadership and collaboration and with highly-skilled and flexible employees are conducive to innovation and thus are best adapted to Agile methodologies. Agile projects work best with a style of management that favors leadership-and-collaboration where the project manager is more of a facilitator or coordinator than a commander, with a desire to put functional code in the hands of the customer earlier, with highly-skilled team members that favor sharing tacit knowledge over documentation and where the value of the product to be delivered is very important to customers. The authors suggest that the types of controls, roles and skills among team members and team communication are different in Agile versus Waterfall projects. Waterfall works best in projects with developers that work on the entire solution as one large deliverable, customers that have minimal involvement after the initial specifications of their needs, and communication that occurs primarily through formalized documentation that codifies both process and product knowledge.

The authors articulate the strengths of each ideology and offer that some experts suggest finding the right balance for a particular organization in implementing aspects of both ideologies rather than claiming either one or the other is supreme. The authors note that the strengths of Waterfall include meticulous and extensive planning, explicit knowledge management, specialized roles, and formal communication. The authors identify the strengths of Agile as an ability to deal with unpredictability by relying on people over processes, empowering team members with more discretionary and decision-making powers so they can self-organize and better respond to emergent situations, and promoting thought-action-reflection activities so the team can continuously learn and adapt.

The authors include a succinct list of characteristics for each of the Waterfall (traditional) and Agile ideologies. Waterfall characteristics include strict planning and control throughout the phases of the PLC accomplished by a command and control style of management, formalized documentation, low customer involvement, and development of the entire solution in one phased approach that occurs after planning and before testing. Agile characteristics include “short iterative cycles of development driven by product features, periods of reflection and introspection, collaborative decision making, incorporation of rapid feedback and change, and continuous integration of code changes into the system under development" (p. 75). Features are decided on jointly by customers and the project team before being implemented in each Agile development cycle.

The authors conclude that while Agile methodologies offer benefits and opportunities, organizations should carefully consider if and how they integrate these methodologies with their existing practices. The authors' work will be useful in establishing a solution to the problem statement described within this literature review wherein I can provide a set of characteristics that will become the criteria for decision-makers to understand when and how to incorporate various techniques of Agile methodologies.

Best Practices in Implementing a Hybrid Approach

Dingsøyr, T., Dybå, T., Gjertsen, M., Odgaard Jacobsen, A., Mathisen, T., Ole Nordfjord, J.,…Strand, K. (2018). Key lessons from tailoring agile methods for large-scale software development. *IEEE IT Professional*. Retrieved from https://arxiv.org/abs/1802.05118

Abstract: We describe advice derived from one of the largest development programs in Norway, where twelve Scrum teams combined agile practices with traditional project management. The Perform program delivered 12 releases over a four-year period, and finished on budget and on time. In this article, we summarize 12 key lessons on five crucial topics, relevant to other large development projects seeking to combine Scrum with traditional project management.

Summary: This source is relevant for this study because it reviews lessons from practitioners who tailored Agile methods to large-scale projects. The authors note that while Agile methods were originally thought to be best suited for small development teams developing software projects that were not life critical, the popularity of Agile has contributed to its adoption in larger projects. They explain that large projects tend to run a higher risk of overrunning the planned scope, schedule, and budget. The authors note that frameworks intended for managing large-scale Agile projects have begun to appear, including Scaled Agile Framework (SAFe) and Large-Scale Scrum (LeSS). They explain that there are few studies focused on these frameworks, and that the frameworks focus on product development while many organizations may be implementing them for projects or entire programs whose focus is developing new systems. The authors note that projects differ from programs in that their timeframe is more limited and they usually have project participants who need to learn the new domain.

The authors studied a large Norwegian development project, called the Perform Program, where twelve Scrum teams combined Agile and traditional project management methods. The authors note that the program successfully delivered twelve releases over a four-year period and finished on time and on budget. The authors summarize advice gleaned from the Perform Program that they believe is relevant to other large development projects seeking to combine Agile and traditional project management methods, categorized as follows: managing the product backlog, which is a prioritized queue of all of the work within the project; ensuring high quality user stories that effectively describe the solution to the developers; facilitating high levels of coordination between the technical and business team members; promoting high quality releases through testing and validation; and ensuring continuous improvement through learning, which is done at the end of each sprint and release in the form of a retrospective ceremony. The authors' research into, and proffered advice from, the key lessons from tailoring and scaling Agile methods will help address the Problem Statement described in this study.

Karlström, D. & Runeson, P. (2005). Combining agile methods with stage-gate project management. *IEEE Software, 22*(3), 43-49. https://doi.org/doi:10.1109/MS.2005.59

Abstract. We encountered cultural changes and different management and engineering viewpoints in a study of three large software development organizations that started using agile methods. Case studies at three large software product companies show the benefits and pitfalls of integrating agile methods with stage-gate management models.

Summary: The authors of this journal article present the results of three case studies of three companies that started using Agile ideology. The authors used a strict qualitative approach to investigate these companies that integrated specific Agile teams into the existing stage-gate software product development. The authors approached this qualitative research as a means of building upon previous work in this field. The authors collected data through their case studies, then analyzed and interpreted the data and presented their findings.

The authors summarized their findings, within the context of their three case studies, in four categories: planning and prioritization, communication and follow-up, process models and roles, and project management. For planning and prioritization, the authors suggest that Agile development implies that the most important features are being worked on at any given time and are completed first. The authors further suggest that this establishes feedback loops that prevent the project deadline from impacting their scope of development work. The authors imply that this prioritization means that less critical features are the only ones that may be scaled back or dropped entirely.

For follow-up and communication, the authors explain that progress is easily demonstrated to customers on Agile projects through project demos as functional software is delivered incrementally. The authors contend that management and developers will both be able to recognize project status clearly with less investment in estimating progress. For process models and roles, the authors recognize the complexity of the customer role in large software development projects and note that the level of representation by customers varied during different intervals in the projects they studied. The authors argue that a project requires strong customer representation linked to the developers, such as a Business or Systems Analyst who represents the requirements of the customer to the developers, as well as an overall decision-maker who is ultimately responsible for leading and coordinating the overall effort, such as a project manager.

For project management, the authors note the need to elicit increased support throughout the organization in order to avoid rejection at the management level. The authors contend that increasing support can happen by offering training that is appropriate to various company levels and functions or by establishing forums for individuals to contribute their own ideas for proposed changes.

The authors' analysis contributes a reference list of current research that provides additional avenues for me to explore, including one that focuses on integrating the Agile and Waterfall ideologies and one that focuses on large corporations. Both of these avenues will be helpful in framing a solution to the problem statement included in this literature review.

Kuhrmann, M., Diebold, P., Munch, J., Tell, P., Garousi, V., Felderer, M.,…Prause, C. (2017). Hybrid software and system development in practice: Waterfall, scrum, and beyond. *ICSSP 2017 Proceedings of the 2017 International Conference on Software and System Process, (30-39).* Paris, France: ACM. https://doi.org/10.1145/3084100.3084104

Abstract: Software and system development faces numerous challenges of rapidly changing markets. To address such challenges, companies and projects design and adopt specific development approaches by combining well-structured comprehensive methods and flexible agile practices. Yet, the number of methods and practices is large, and available studies argue that the actual process composition is carried out in a fairly ad-hoc manner. The present paper reports on a survey on hybrid software development approaches. We study which approaches are used in practice, how different approaches are combined, and what contextual factors influence the use and combination of hybrid software development approaches. Our results from 69 study participants show a variety of development approaches used and combined in practice. We show that most combinations follow a pattern in which a traditional process model serves as framework in which several fine-grained (agile) practices are plugged in. We further show that hybrid software development approaches are independent from the company size and external triggers. We conclude that such approaches are the results of a natural process evolution, which is mainly driven by experience, learning, and pragmatism.

Summary: This source is relevant for this study because the authors propose to study the various software and system development approaches in practice, the ways that combinations of approaches are being implemented, and the triggers for creating a hybrid approach. The authors find a gap in the literature wherein Agile research is rich and traditional processes have been mostly ignored in recent research. The authors suggest this disparity results in a lack of information on the factors driving the combination of approaches to create hybrid approaches, such as how industry standards impact the use of Agile and whether company size matters.

The authors found that hybrid approaches are widely used by companies of varying size, within a range of industry sectors. The authors found no evidence to suggest that the size of an organization is a trigger or a barrier to using hybrid methods. The authors assert that companies adopt Agile methods despite the challenges they face from standards and regulations, such as ISO 9001, which is the international standard that specifies requirements for a quality management system; and ISO 27001, which is a specification for an information security management system. The authors note that hybrid approaches generally use a traditional approach as the initial framework, which is then refined by adding Agile components. The authors find that the customization of project processes is mostly achieved by finely tailoring individual practices to suit the organization.

The authors report that the primary limitation of their study is the population because most of the participants were located in Europe. Additionally, the authors state that convenience sampling also imposes limitations on the study because less control is applied to the distribution of participants across sector or job roles. The authors note that this study has limited generalizability but does improve their understanding of approaches in practice. The authors' research into hybrid development practices will help frame a solution for the Problem Statement described in this study.

Papadakis, E. & Tsiropis, L. (2018). Hybrid methods and practices associated with agile methods, method tailoring and delivery of projects in a non-software context. *Procedia Computer Science, 138*, 739-746. https://doi.org/10.1016/j.procs.2018.10.097

Abstract: Nowadays the business world is characterized by complexity since market environment is changing quickly. Delivery practice and methods for project, program and portfolio management have changed over the decades to align themselves to the changing environment. Unlike traditional methods used in delivery of projects and programs, agile methods are marked by responding to change over following a plan and by extensive collaboration with customer over contract negotiation and offering a variety of benefits that make them attractive to researchers. Although the latter methods [1] claimed to be beneficial presenting advantages such as accelerate time to market, increase in quality and productivity, new trends and hybrid methods and tailored processes are being discussed and developed. In this study the authors provide a report analysis of proposed tested frameworks already presented in the relative literature, tailored methodologies, a review of most used and popular agile practices and approaches and the trends in our subject area conducting a literature review ranging from 2000 to 2017. Our research strategy following a systematic approach [2] revealed 524 studies, of which 71 had been identified to answer our research questions. This is part of further work based on the first authors' PhD work. The results will be a guide to choose the most appropriate blend of practices for a given project, adapt them to the changing needs and develop an innovative framework methodology.

Summary: This source is relevant for this study because the authors explore emerging hybrid methodologies which are frameworks incorporating popular Agile practices. The authors note that Agile methods are attractive because they promote a response to change and collaboration with customers. The authors state that in spite of the benefits of Agile, such as an increase in both productivity and quality, hybrid approaches are being discussed and developed among practitioners.

The authors define hybrid as a combination of two different project management methodologies, Agile and traditional, in an effort to create a new, improved project management model. The authors contend that hybrid methodologies can nimbly adapt to the fluidity of projects and allow for a more nuanced approach to the work because Agile techniques can be applied to varying degrees depending on the needs of the project. The authors explain that having more ways of approaching a problem increases the project team's chance of resolving problems. The authors note that developing a hybrid methodology by combining elements from both Agile and traditional methodologies can provide multiple approaches to project problems and enable practitioners to take advantage of the strengths of each and mitigate the risks inherent in their weaknesses.

The authors found current research that is revealing tailored methods that will better prepare organizations for innovation. The authors report that the main benefits emerging from these hybrid methods were related to productivity, flexibility, teamwork, and customer involvement. In examining applications of Agile tailored and hybrid methodologies outside of software development, the authors discovered that projects delivering services provide the most appropriate context for these processes. The authors assert that the results of their literature review are contributing to their development of an

Agile hybrid model for services-provisioning projects that will change the future of value delivery, creating value for an organization through strategic advantage.

The authors capture the current state of the identified project methodologies: Agile, traditional, and hybrid, paying special attention to those focused on domains other than software development. The authors find that the research includes discussions on both adoption and adaptation procedures. The authors point out that scope, schedule, and budget, while still essential, are no longer sufficient as the primary measures of project success. The authors find that relevant measures of success include how well the project adapts to changing customer requirements, how productive the development team is, how collaborative and communicative the project team is, and how fast they can get to market. The authors contend that the benefits of a hybrid approach include the ability to respond faster and more flexibly to change, better integration of customer involvement, improved communication, better productivity by developers, and decreased time to market. The authors' research into the usage of hybrid methodologies in non-software domains will help frame a solution to the Problem Statement described in this study.

Rahimian, V., & Ramsin, R. (2008). Designing an agile methodology for mobile software development: A hybrid method engineering approach. *2008 Second International Conference on Research Challenges in Information Science, Marrakech, 2008, (pp. 337-342)*. https://doi.org/10.1109/RCIS.2008.4632123

Abstract: New Advances in mobile computer technology and the rapid growth of wireless networks in quality and quantity has introduced new applications and concerns in computer science and industry. The unique requirements and constraints associated with mobile systems have brought new challenges to software development for such environments, as it demands extensive improvements to traditional systems development methodologies in order to fulfill the special needs of this field. We examine the challenges of developing software for mobile systems, starting by reviewing mobile systems' characteristics and investigating the status quo of mobile software development methods. It has been shown that Agile methodologies are appropriate methods for the development of such systems; based on this assumption, we identify specific requirements for a mobile software development methodology, based on which a new agile method is engineered using the Hybrid Methodology Design approach. We claim that this methodology, and the approach used for its construction, can facilitate the application of a software engineering approach to the production of mobile software systems.

Summary: This source is relevant for this study because the authors describe the characteristics of an ideal mobile software development methodology using a hybrid methodology design approach. The authors note that Agile methodologies enhance both flexibility and productivity because they adapt to changes in both requirements and the environment and promote learning from past experiences. The authors say that the Agile characteristics that are the most important for successful software development are iterative processes, test-driven development, the use of highly skilled developers, and continuous improvement through user reviews.

The authors describe utilizing a hybrid approach, using top-down, incremental processes, that includes the characteristics they outlined and is based on their professional experience as well as knowledge they gained from researching existing software development methodologies. They note that a primary process included in the approach is the prioritization of the requirements, which is done at the beginning of the project and again at the end of each iteration. The authors suggest that to design the correct hybrid mix of project management methodologies one should do the following: select the design engine (instantiating an existing model, devising an artifact-oriented chain and building the process around it, composing a process around an existing pattern, or integrating features and techniques from existing methodologies), apply the selected design to define the methodology according to current scope, iteratively refine and revise the methodology to accommodate changes throughout the life-cycle, specify the scope of each iteration, and revise and reprioritize the requirements at each iteration. The authors contend that the set of prioritized requirements should focus on designing the most important elements first, and then integrating them into the overall product blueprint.

The authors conclude that an optimal hybrid approach for mobile software development is an Agile risk-based methodology that is highly influenced by a particular formal mobile software development framework called New Product Development (NPD), which focuses on solution requirements and validates results against those requirements. The authors' research into the effectiveness of hybrid methodologies will help frame a solution for the Problem Statement described in this study.

Schmitz, K., Mahapatra, R. K., & Nerur, S. (2018). User engagement in the era of hybrid agile methodology. *IEEE Software,* 1. https://doi.org/10.1109/MS.2018.290100623

Abstract: Contemporary software development and implementation projects are increasingly adopting agile methods by tailoring and blending agile techniques into a traditional project framework (and sometimes vice versa). Common tailoring methods employed by project teams emphasize flexibility to embrace local project context but risk neglecting important communication dynamics necessary for effective information exchange. This study of two hybrid IS projects highlights the distinction between frequent participation and effective engagement. Lessons from these projects suggest tailoring should remove developer orchestration from user feedback activities in order to expose hard to find requirements and replace pseudo-real scenarios with actual-real situations for effective user training. These recommended tailoring constraints restore balance to the information exchange process to improve the communication coupling between project teams and users.

Summary: This source is relevant for this study because the authors explore how communication in the project team is affected by hybrid methods. The authors contend that project communications should focus on eliciting engagement. The authors note three levels of interaction, which they broke into categories: participation, involvement, and engagement. The authors note that participation is behavioral and the main focus for communications is the frequency in which they are delivered. The authors state that involvement is motivational, so the focus of communications should be to characterize the significance of the project outcome as well as the activities taking place to achieve the intended outcome. The authors note that motivating users may be more difficult than motivating the project team because users are often tied to the legacy system and are still using it daily, so their view is constantly informed by the current state. The authors suggest that engagement is also behavioral, but it promotes meaningful exchanges of information through high quality communication and is the desired result of motivated involvement from all users and project team members. The authors find that it is important for teams using hybrid methods to seek high levels of engagement to work together effectively and develop a deep understanding of project requirements and solution descriptions.

The authors note that processes are required to manage any existing gaps between the project vision and the motivation of project stakeholders to achieve effective engagement and facilitate a well-balanced exchange of information. The authors say that without this balance between participation and involvement, the project team can easily dominate communication, which leads to a decrease in the understanding needed for a successful project. The authors find that rather than having a developer-led feedback session, promoting user-guided sessions will help reveal harder to find requirements. The authors also find that using real situations to the maximum extent possible in training scenarios will increase the probability for success.

The authors conclude that communications are artificial and controlled in a project, which can lead to an imbalance between participation and involvement and therefore become a barrier to effective engagement. The authors suggest that the most obvious sign of imbalance is a lack of understanding; either a user will fail to gain an understanding of the proposed solution or a team member will fail to gain an understanding of the requirements that are most critical to the users. The authors conclude that paying attention to the mechanisms for communication within a project can provide lessons on the effectiveness of the hybrid approach. The authors' research into communication styles of teams using hybrid methods will help frame a solution for the Problem Statement described in this study.

Špundak, M. (2014). Mixed agile/traditional project management methodology—Reality or illusion? *Procedia - Social and Behavioral Sciences 119,* 939 – 948. https://doi.org/10.1016/j.sbspro.2014.03.105

Abstract: Project management methodology is usually defined as a set of methods, techniques, procedures, rules, templates, and best practices used on a project. It is commonly based on a specific project management approach, that defines a set of principles and guidelines which define the way a project is managed. With the growing trend of usage of agile project management on different projects, it is clear that two opposite sides exist – traditional and agile project management approach, and that there exists a need to combine both approaches. So, the question is if it is and how it is possible to combine both approaches in a single project management methodology? The paper covers thorough literature review and starts with the definition of the project management approach and of the project management methodology. It provides overview of different project management approaches and defines project management methodologies. The literature review shows what is considered as part of project management methodology in a wider or narrower sense, and what the main characteristics of a methodology are. The need for combining project management approaches is shown on the case of software development project. The paper provides basis for further research on application of different project management approaches and methodologies. Further research could build on an idea of creating unique methodology for project, based on different project management approaches. In that way it is possible to create project management methodologies that have high possibility of customization to projects and to project environments.

Summary: This source is relevant for this study because it defines project management and explores how Agile project management differs from traditional or Waterfall project management. The author's main question is: "Is it possible to combine different approaches within a single project management methodology?" (p. 940). The author also asks if there is any best methodology that can be labeled the most optimal for a particular environment, such as company, organization, or division, or if an adaptation of a methodology can create a best fit for a particular environment.

The author contends that project management involves tailoring and applying methodologies, rules, processes, and templates to a specific organization and project. The author notes traditional project management is the prevailing approach in knowledge produced by professional project management organizations. The author suggests that the goal of traditional project management is to complete the project within the scope, schedule, and budget that was initially planned. The author notes that "even though traditional approach to project management emphasizes robustness as one of its advantages, prescribing that the same methods and techniques could be applied to all projects uniformly, it is increasingly mentioned as one of the crucial disadvantages of such approach" (p. 941). The author asserts that, as business environments are increasingly complex, change is inevitable and adaptability is necessary to manage change, but traditional project management is based on predictability.

The author notes that Agile project management is the relatively new approach to project management and is strongly tied to software development projects. The author reports that all Agile methodologies share a common characteristic of being adaptable to change during the project PLC and customizable to various projects. The author finds that

Agile approaches emphasize project execution over planning, and communication and collaboration over pure process following. The author notes that these differences may require an organization to change their way of thinking, suggesting that practicing an Agile approach means having the ability to balance flexibility with stability. The author notes that critical success factors for Agile include having highly skilled project team members and using the best delivery strategy through the appropriate sizing of functional releases. The author finds that the largest challenges come from organizational issues such as disagreement over which methodology to adopt, less experienced team members, lack of full-time dedicated team members, and infrequent contact between project manager and team members.

The author stresses that each challenge can be avoided by recognizing and understanding the differences between the Agile and Waterfall approaches, creating organizational culture that is prepared to embrace change, and through patience and hard work. The author notes that the benefits of a project management methodology can be demonstrated in many specific ways, such as reduced risks, faster time to market, better control over scope creep, improved decision-making processes, and better knowledge management.

In differentiating Agile and Waterfall project management methods, the author notes that the traditional approach is more appropriate for projects that have clear user requirements up front and clearly identified project goals which also means they have low levels of uncertainty. The author notes that Waterfall methods will likely be successful in these projects due to the low rate of change expected in these projects along with less need for user involvement. The author says that these projects will follow a linear and predictable path based on the initial plan and the focus will be on optimizing the efficiency of the execution of project activities. The author finds that the traditional approach is also appropriate when formal documentation is a strict requirement and operations are routine and predictable such as construction and engineering projects.

On the other hand, the author notes that Agile project management is better suited for creative, innovative projects, such as new innovative product development projects. The author explains that these projects share similar characteristics such as high level of uncertainty, unclear project goals and incomplete up front requirements. The author notes that iterative Agile methods will likely be successful in these projects due to the frequent change orders, need for close collaboration, the feedback loop that focuses the customer on their true requirements, and frequent updates to the project plan. The author notes that an iterative Agile approach is beneficial when fast implementation is required due to time constraints.

The author contends that the goal of choosing the optimal hybrid approach is to increase the likelihood for successful project completion. The author warns that a methodology is not sufficient for project success if the project team does not fully understand the scope of the work required. The author suggests that an inappropriate choice or application of a methodology can negatively impact the project and inhibit success. The author explores the characteristics that they recommend must align in order to increase the likelihood of success in a project, such as alignment with organizational processes, understanding limitations of the methodology, and alignment within the organization and with customer organizations.

The author finds that further literature review will be necessary in order to answer their initial research questions. The author identifies the challenge of defining a hybrid project management methodology, which is highly customizable within the project and organizational parameters but is highly dependent on accurately identifying which project characteristics inform the customization decisions. The author further notes that advantages and disadvantages exist for both Agile and traditional approaches. The author suggests that selecting an approach takes care and consideration of the organization, and that it is possible to use a hybrid approach if the practitioner knows when to lean on each methodology included in the hybrid. The author warns that a methodology needs to be adapted to a project, not the other way around. The author's research into the characteristics that must be considered when selecting project management methodologies will help frame a solution for the Problem Statement described in this study.

Conclusion

This study seeks to present literature to assist the reader in understanding the best practices in implementing a hybrid project management methodology that will align stakeholders with differing project approaches. Sources are presented that provide a historical context for Waterfall and Agile methodologies, as well as sources that describe best practices in and challenges to avoid with hybrid project management implementations. Project Managers, Directors and other leaders who need a deeper understanding of the challenges that face project teams when implementing a hybrid project management approach will benefit from the research provided in this annotated bibliography.

Characteristics of Waterfall Projects

Characteristics of Waterfall. The Waterfall model for project management was first introduced in 1970 by Royce (Bassil, 2012; Larman & Basili, 2003). Bassil (2012) defines the Waterfall model as “a sequential software development process in which progress is regarded as flowing increasingly downwards (similar to a waterfall) through a list of phases that must be executed in order to successfully build a computer software” (p. 2). Bassil (2012) defines each of the five phases of the Waterfall model: (a) analysis phase, where functional requirements are defined, (b) design phase, where components such as software architecture and data structure are designed, (c) implementation phase, where a usable tool is realized, (d) testing phase, where verification and validation occur, and (e) maintenance phase, where issues are corrected and enhancements are implemented. The primary tools and techniques of Waterfall or traditional project management methods are the work breakdown structure (WBS), which provides the basic definition of the project work; responsibility matrix, which assigns project team members to the work listed in the WBS; Gantt chart, which provides a visual representation of the project schedule; project network techniques, which illustrate scheduling impacts that activities have on each other and provide the means to determine critical activities and float times; cost scheduling, which identifies the necessary capital investment; and a project control process, which will detect project overruns in schedule or budget (Rodrigues & Bowers, 1996). Waterfall projects share characteristics such as strict planning and control throughout the phases of the PLC accomplished by a command and control style of management, formalized documentation, low-customer involvement, and development of the entire solution in one phased approach that occurs after planning and before testing (Nerur et al., 2005).

Projects where Waterfall works best. Špundak (2014) notes that the traditional approach is more appropriate for projects that have clearly defined user requirements at the beginning of the project and clearly identified project goals, which also means they have low levels of uncertainty. Waterfall methods will likely be successful in these projects due to the low rate of change expected and less need for user involvement (Špundak, 2014). These projects will follow a linear and predictable path based on the initial plan and the focus will be on optimizing the efficiency of the execution of project activities (Špundak, 2014).

The traditional approach is also optimal when formal documentation is a strict requirement and operations are routine and predictable, such as in construction and engineering projects (Špundak, 2014). The classic control cycle and well-ordered project system of traditional project management may not be optimal for design and development projects that require a more iterative nature (Rodrigues & Bowers, 1996). The tactical-level approach of traditional techniques work well for operational issues that have clearly defined objectives and constraints (Rodrigues & Bowers, 1996). Waterfall works best in projects with developers that work on the entire solution as one large deliverable, customers that have minimal involvement after the initial specification of their needs, and communication that occurs primarily through formalized documentation that codifies both process and product knowledge (Nerur et al., 2005).

Benefits of Waterfall. The benefits of employing a Waterfall approach to project management include the ability to estimate all of the work required to complete the project, assess costs of and resources necessary for that work, and control the costs and work throughout the project life-cycle (Rodrigues & Bowers, 1996). The integration of subject matter experts from the organization into the project team and the clearly defined work structure allow for continuous control and planning activities as the implementation proceeds (Rodrigues & Bowers, 1996). The strengths of Waterfall include meticulous and extensive planning, explicit knowledge management, specialized roles, and formal communication (Nerur et al., 2005).

Challenges with Waterfall. Challenges with a Waterfall approach include its inflexibility towards change (Boehm & Turner, 2005; Ciric et al., 2018; Petersen et al., 2009; Rahimian & Ramsin, 2007; Rodrigues & Bowers, 1996), the amount of rework on a typical Waterfall project and unpredictable software quality due to the delay of testing until late in the project (Petersen et al., 2009). Traditional techniques concentrate on detailed planning, which encourages a narrow, operational view of the project (Rodrigues & Bowers, 1996). Most traditional project management techniques focus on tactical-level approaches, but strategic-level approaches are necessary due to the increasingly complex and critical nature of projects (Rodrigues & Bowers, 1996).

Overall, the most critical challenges relate to the gathering of requirements in the analysis phase when too much documentation is created that is never used later in the project, and not thoroughly assessing the full impact of requirements (Petersen et al., 2009). Verification that occurs in the testing phase can be challenging if schedule overruns limit the testing time left at the end of the project (Petersen et al., 2009). Defects that are discovered late are more difficult and expensive to fix (Petersen et al., 2009).

Characteristics of Agile Projects

Characteristics of Agile. Ciric et al. (2018) report that Agile project management arose in response to the inadequacy of traditional methods in adapting to increasing complexity within projects that also experience changing goals and uncertainty around outcomes. Prominent practitioners of software development gathered in 2001 to identify methods for the software development industry to achieve better results and overcome limitations of traditional processes (Ciric et al., 2018). Multiple authors note that this group created the Manifesto for Agile Software Development, which contained principles and values expressing the basic ideology of Agile software development (Beck et al., 2001; Ciric et al., 2018). The four values emphasized individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan (Beck et al., 2001).

The most prominent characteristics of Agile projects are the incremental nature of the development work, the iterative nature of requirement prioritization, and the close collaboration of the project team, which includes project management personnel, developers, and customers (Beck et al., 2001; Boehm & Turner, 2005; Ciric et al., 2018; Karlström & Runeson, 2005; Kuhrmann, 2017; Nerur et al., 2005; Schmitz et al., 2018; Špundak, 2014). Development work is completed by self-organizing teams, in pre-determined increments of time (e.g., two weeks) called sprints, and each sprint is reviewed with a retrospective learning session (Nerur et al., 2005; Papadakis & Tsironis, 2018).

Agile approaches emphasize project execution over planning, and communication and collaboration over pure process following (Špundak, 2014). Agile approaches also emphasize collaborative decision making and continuous incorporation of feedback and change (Nerur et al., 2005). These differences may require an organization to change their way of thinking, suggesting that practicing an Agile approach means having the ability to balance flexibility with stability (Špundak, 2014). Špundak (2014) lists critical success factors for Agile as having highly skilled project team members and using the best delivery strategy through the appropriate sizing of functional releases. Features are decided on jointly by customers and the project team before being implemented in each development cycle (Nerur et al., 2005). The Agile characteristics that are the most important for successful mobile software development are the use of iterative processes, test-driven development, highly skilled developers, and the pursuit of continuous improvement through user reviews (Rahimian & Ramsin, 2008).

Projects where Agile works best. Agile project management is better suited for creative, innovative projects, such as new innovative product development projects (Špundak, 2014). These projects share similar characteristics such as high levels of uncertainty, unclear project goals and incomplete up-front requirements (Špundak, 2014). Iterative Agile methods will likely be successful in these projects due to the frequency of changes, need for close collaboration, feedback loops that focus the customers on their true requirements, and frequent updates to the project plan (Špundak, 2014). An iterative Agile approach is beneficial when fast implementation is required due to time constraints (Špundak, 2014).

Benefits of Agile. Nerur et al. (2005) report that teams using Agile will experience increased happiness as a result of its reliance on people and their creativity to deal with unpredictability. These empowered team members successfully address project challenges, which leads to increased satisfaction (Nerur et al., 2005). Rahimian and Ramsin (2008) note that Agile methodologies enhance both flexibility and productivity because they adapt to changes in both requirements and the environment and promote learning from past experiences. Ciric et al. (2018) list benefits of APM techniques as greater flexibility, reduced planning time, reduced cost and schedule, delivery of what the customer expected, and improved communications. In one case study, Circa et al. (2018) found that Agile methods improved the practices within the organization studied regarding how the organization transfers knowledge and understanding of the needs of their customers that can be applied to innovation.

Agile seeks to respond to change rather than viewing it as a constraint; customer feedback is therefore incorporated and improves the solution being delivered (Boehm & Turner, 2005). Changing requirements are seen as opportunities to meet the customer’s needs rather than an obstruction to project success (Boehm & Turner, 2005). Delivering working software in each release is a better measure of progress and detects, and thus prevents low-value functionality earlier in the project (Boehm & Turner, 2005). Agile has the ability to deal with unpredictability by relying on people over processes, empowers team members with more discretionary and decision-making powers so they can self-organize and better respond to emergent situations, and promotes thought-action-reflection activities so the team can continuously learn and adapt (Nerur et al., 2005).

Challenges with Agile. Primary challenges for an organization that is implementing Agile methodologies are letting go of their process-driven approaches and adopting a feature-driven, people-centric approach; embracing iterative, test-driven development that emphasizes adaptability; and managing large, scalable projects (Nerur et al., 2005). Project managers also face challenges such as letting go of the authority that comes with a command and control style and embracing their role as facilitator/coordinator in an Agile approach (Nerur et al., 2005). An organization's reliance on existing technology may hinder attempts to adopt Agile methodologies because Agile requires different tools, which are critical to Agile project success (Nerur et al., 2005).

Challenges can occur while adapting to new lightweight development processes; restructuring development teams into medium-sized teams, configured into a team of teams; and ensuring each team has a competent team leader possessing an appropriate mix of technical, people, and Agile skills (Boehm & Turner, 2005). New challenges are posed when teams have to work with shorter life cycles (Boehm & Turner, 2005). Teams also struggle with replacing or enhancing legacy systems developed using Waterfall methods because they are not generally easy to refactor to accommodate Agile replacements or enhancements that build capability incrementally (Boehm & Turner, 2005).

Business process challenges arise when an organizational culture requires accurate predictions of tasks and does not encourage experimentation or evolution (Boehm & Turner, 2005). An organization must assess the required skills and level of experience of team members while enabling a culture wherein rewards are based on team achievement rather than individual achievement (Boehm & Turner, 2005). Changing the processes for measuring progress can be difficult as teams use requirements burndown to measure the quantity of requirements from the backlog that have been marked completed rather than traditional milestone reviews (Boehm & Turner, 2005). Challenges may stem from a lack of skills because Agile methods require an ability to task shift with ease, or there may be logistical issues that arise from a lack of colocation (Boehm & Turner, 2005). If the culture does not currently encourage trying new things, an organization may face negative emotions that stem from feelings of inadequacy, jealousy, and defensiveness, which can lead to undesirable behaviors that may undermine the new methods (Boehm & Turner, 2005).

Best Practices in Implementing a Hybrid Approach

Hybrid project management borrows, mixes, and blends techniques from Agile and Waterfall methodologies in order to tailor a project-specific methodology (Papadakis & Tsiropis, 2018; Schmitz et al., 2018). Hybrid project management arose out of the desire to realize the benefits of Agile methodologies while still maintaining some Waterfall structure (Špundak, 2014). The customized, project-specific combination of methodologies is a benefit of the hybrid approach that requires flexibility on the part of the project team (Schmitz et al., 2018) and will better prepare organizations for innovation (Papadakis & Tsiropis, 2018).

Gain organizational support. The project team needs to elicit increased support throughout the organization in order to avoid rejection at the management level (Karlström & Runeson, 2005). Increased support can be gained by offering training that is appropriate to various company levels and functions or by establishing forums for individuals to contribute their own ideas for proposed changes (Karlström & Runeson, 2005). One of the twelve principles of the Agile Manifesto for Software Development is "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done" (Beck et al., 2001, para. 1).

Tailoring a hybrid approach. Kuhrmann et al. (2017) find that hybrid approaches generally use a traditional approach as the initial framework, which is then refined by adding Agile components. For the specific case of developing applications for mobile systems, Rahimian and Ramsin (2008) conclude that an optimal hybrid approach design includes an Agile focus on solution requirements and then validates the results against those requirements.

Designing the correct hybrid mix for an organization includes selecting the design engine by instantiating an existing model, devising an artifact-oriented chain and building the process around it, composing a process around an existing pattern, or integrating features and techniques from existing methodologies; applying the selected design to define the methodology according to the current scope; iteratively refining and revising the methodology to accommodate changes throughout the life-cycle; specifying the scope of each iteration; and revising and reprioritizing the requirements at each iteration (Rahimian & Ramsin, 2008).

When selecting an approach, the practitioner must know when to lean on each methodology they choose to include in the hybrid (Špundak, 2014). The traditional approach is more appropriate for projects that have clear user requirements up front and clearly identified project goals, which also means they have low levels of uncertainty (Špundak, 2014). Waterfall methods will likely be successful in these projects due to the low rate of change expected in these projects, along with less need for user involvement (Špundak, 2014). These projects will follow a linear and predictable path based on the initial plan and the focus will be on optimizing the efficiency of the execution of project activities (Špundak, 2014). The traditional approach is also appropriate when formal documentation is a strict requirement and operations are routine and predictable, such as construction and engineering projects (Špundak, 2014).

On the other hand, Agile project management is better suited for creative, innovative projects, such as new innovative product development projects (Špundak, 2014). These projects share similar characteristics such as high levels of uncertainty, unclear project goals and incomplete up-front requirements (Špundak, 2014). Iterative Agile methods will likely be successful in these projects due to frequent change orders, the need for close collaboration, the feedback loop that focuses the customers on their true requirements, and frequent updates to the project plan (Špundak, 2014). An iterative Agile approach is beneficial when fast implementation is required due to time constraints (Špundak, 2014).

Collaborate closely with the development team. Dingsøyr et al. (2018) explain the importance of closely collaborating with the development team to ensure that the combination of Agile and traditional project management methods is achieving success. The primary collaboration concerns for a project team working with the developers are categorized as follows: managing the product backlog, which is a prioritized queue of all of the work within the project; ensuring high quality user stories that effectively describe the solution to the developers; facilitating high levels of coordination between the technical and business team members; promoting high quality releases through testing and validation; and ensuring continuous improvement through learning, which is done at the end of each sprint and release in the form of a retrospective ceremony (Dingsøyr et al., 2018).

Communicate. Project progress on hybrid projects is easily communicated through product demonstrations to customers as functional pieces of software are delivered incrementally (Karlström & Runeson, 2005). An added benefit of this approach is that project managers can clearly see progress with less of their own time invested in estimating project status (Karlström & Runeson, 2005).

The controlled communications that are part of traditional project management approaches tend to be artificial and can become a barrier to effective engagement due to an imbalance between participation and involvement (Schmitz et al., 2018). The most obvious sign of this type of imbalance is a lack of understanding; either a user will fail to gain an understanding of the proposed solution or a team member will fail to gain an understanding of the requirements that are most critical to the users (Schmitz et al., 2018)**.** Paying attention to the mechanisms for communication within a hybrid project can provide lessons on the effectiveness of the hybrid approach, which can drive continuous improvement (Schmitz et al., 2018). The goal of project communication in hybrid methodologies should be to seek high levels of engagement that promote meaningful exchanges of information that build a deep understanding of project requirements and solution descriptions (Schmitz et al., 2018).

Plan and prioritize. Hybrid methodologies should embrace the Agile practice of prioritizing requirements, working on the most important ones first (Karlström & Runeson, 2005; Rahimian & Ramsin, 2008). Having regular planning and prioritizing review sessions establishes feedback loops that are critical to project success and ensures that less critical features are the only ones that may be scaled back or dropped entirely (Karlström & Runeson, 2005; Špundak, 2014).

Practices to avoid when implementing hybrid approaches. A project team should not select a hybrid mix without fully understanding the scope of work required because a methodology alone is not sufficient for project success (Špundak, 2014). A methodology needs to be adapted to a project, not the other way around (Špundak, 2014). A hybrid project management methodology is highly customizable within the project and organizational parameters but is highly dependent on accurately identifying which project characteristics inform the customization decisions (Špundak, 2014).

The project team should not dominate communication, which leads to a decrease in the understanding needed for a successful project (Schmitz et al., 2018). The feedback sessions should not be developer-led since this may close off customer interaction and will not effectively draw out harder to identify requirements or defects (Schmitz et al., 2018). Business and systems analysts should be cautious of the amount of documentation work they put into business processes and product features because it will be put aside if priorities shift and they will need to prepare new analyses for the recently committed work (Dingsøyr et al., 2018).

Final Thoughts

Waterfall and Agile methodologies each have benefits and challenges that make them well suited for specific types of projects (Petersen et al., 2009; Špundak, 2014). Increasingly complex projects and a drive towards innovation means organizations will benefit from a flexible approach to implementing a project management methodology that is tailored to the needs of the organization and the project it is managing (Papadakis & Tsiropis, 2018; Schmitz et al., 2018). A tailored approach will borrow, mix, and blend techniques from both Agile and Waterfall methodologies in order to adapt an approach that benefits the project and fits well with the organization (Papadakis & Tsiropis, 2018; Schmitz et al., 2018). Hybrid methods ensure the project realizes the benefits of Agile methodologies while maintaining an appropriate amount of Waterfall structure (Špundak, 2014).

References

Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *Lebanese Association for Computational Sciences (LACSC), International Journal of Engineering & Technology (iJET), 2*(5), 742-749. Retrieved from https://arxiv.org/abs/1205.6904

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M.,…Thomas, D. (2001). The agile manifesto. Retrieved from https://moodle2016-17.ua.es/moodle/pluginfile.php/80324/mod_resource/content/2/agile-manifesto.pdf

Boehm, B. & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software 22*(5), 30-39. https://doi.org/10.1109/MS.2005.129

Chari, K. & Agrawal, M. (2017). Impact of incorrect and new requirements on waterfall software project outcomes. *Empirical Software Engineering 23*(1), 165-185. https://doi.org/10.1007/s10664-017-9506-4

Ciric, D., Lalic, B., Gracanin, D., Palcic, I., & Zivlak, N. (2018). Agile project management in new product development and innovation processes: Challenges and benefits beyond software domain. *IEEE Xplore Digital Library.* https://doi.org/10.1109/TEMS-ISIE.2018.8478461

Dingsøyr, T., Dybå, T., Gjertsen, M., Odgaard Jacobsen, A., Mathisen, T., Ole Nordfjord, J., Røe, K., & Strand, K. (2018). Key lessons from tailoring agile methods for large-scale software development. *IEEE IT Professional*. Retrieved from https://arxiv.org/abs/1802.05118

Dingsøyr, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2017). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering 23*(1), 490-520. https://doi.org/10.1007/s10664-017-9524-2

Hass, K. (2007). The blending of traditional and agile project management. *PM World Today, IX(V)*. Retrieved from http://courses.cecs.anu.edu.au/courses/COMP3120/local_docs/readings/the-blending-of-traditional-and-agile-project-management.pdf

Karlström, D. & Runeson, P. (2005). Combining agile methods with stage-gate project management. *IEEE Software, 22*(3), 43-49. https://doi.org/doi:10.1109/MS.2005.59

Kuhrmann, M., Diebold, P., Munch, J., Tell, P., Garousi, V., Felderer, M.,…Prause, C. (2017). Hybrid software and system development in practice: Waterfall, scrum, and beyond. *ICSSP 2017 Proceedings of the 2017 International Conference on Software and System Process, (30-39).* Paris, France: ACM. https://doi.org/10.1145/3084100.3084104

Larman, C. & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer 36*(6), 47-56. https://doi.org/10.1109/MC.2003.1204375

Lee, S. & Yong, H. (2009). Distributed agile: Project management in a global environment. *Empirical Software Engineering, 15(2), 204-217.* https://doi.org/10.1007/s10664-009-9119-7

McCormick, M. (2012). Waterfall vs. agile methodology. *MPCS 2012*. Retrieved from http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM – Adaptive Complex Enterprises 48*(5), 72-78. https://doi.org/10.1145/1070000/1060712

Papadakis, E. & Tsiropis, L. (2018). Hybrid methods and practices associated with agile methods, method tailoring and delivery of projects in a non-software context. *Procedia Computer Science, 138*, 739-746. https://doi.org/10.1016/j.procs.2018.10.097

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. *International Conference on Product-Focused Software Process Improvement--PROFES 2009,* 386-400. https://doi.org/10.1007/978-3-642-02152-7_29

Rahimian, V., & Ramsin, R. (2008). Designing an agile methodology for mobile software development: A hybrid method engineering approach. *2008 Second International Conference on Research Challenges in Information Science, Marrakech, 2008,* 337-342. https://doi.org/10.1109/RCIS.2008.4632123

Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016a). The secret history of agile innovation. *Harvard Business Review online*. Retrieved from https://hbr.org/2016/04/the-secret-history-of-agile-innovation

Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016b). Embracing agile. *Harvard Business Review Online*. Retrieved from https://hbr.org/2016/05/embracing-agile

Rodrigues, A. & Bowers, J. (1996). System dynamics in project management: A comparative analysis with traditional methods. *System Dynamics Review 12*(2), 121-139. https://doi.org.10.1080/08874417.2009.11646044

Schmitz, K., Mahapatra, R. K., & Nerur, S. (2018). User engagement in the era of hybrid agile methodology. *IEEE Software,* 1. https://doi.org/10.1109/MS.2018.290100623

Serrador, P. & Pinto, J. (2015). Does agile work? – A quantitative analysis of agile project success. *International Journal of Project Management, 33*(5), 1040-1051. https://doi.org/10.1016/j.ijproman.2015.01.006

Sommer, A. F., Dukovska-Popovska, I., & Steger-Jensen, K. (2014). Barriers towards integrated product development – Challenges from a holistic project management perspective. *Science Direct International Journal of Project Management 32*(6), 970-982. https://doi.org/10.1016/j.ijproman.2013.10.013

Špundak, M. (2014). Mixed agile/traditional project management methodology – Reality or illusion? *Procedia - Social and Behavioral Sciences 119,* 939 – 948. https://doi.org/10.1016/j.sbspro.2014.03.105

University of Florida Center for Public Issues Education. (2014). *Evaluating information sources.* Retrieved from http://www.piecenter.com/wpcontent/uploads/2014/08/evaluateinfo.pdf

Verma, A. (2018). Agile project management. *International Conference on Recent Innovations in Science and Engineering*. Gida, Gorakhpur: India. Retrieved from http://data.conferenceworld.in/bit.gkp/96.pdf

Zafar, I., Nazir, A. K., & Abbas, M. (2017). The impact of agile methodology (DSDM) on software project management. *Circulation in Computer Science: International Conference on Engineering, Computing & Information Technology (ICECIT 2017), (pp. 1-6)*. Retrieved from https://pdfs.semanticscholar.org/8437/33664dc56367e0c61a6a854a84b844798c45.pdf