ABSTRACT

Herein, we study knowledge acquisition by project teams using waterfall and agile PM methods to understand the effect from (1) process losses to communication during meeting periods; and, (2) process losses to self-study between meeting periods. We develop an agent-based model and apply virtual experimentation to different process loss conditions. The results suggest that teams using waterfall PM achieve comparable or better outcomes to teams using agile PM from low process losses to communications; while teams using agile PM have better outcomes from low process losses to self-study between meeting periods. Implications for research and practice are discussed.

KEYWORDS: Project management, Communication, Waterfall software development, Agile software development, Agent-based modeling

INTRODUCTION

Organizations often implement Project Management (PM) to facilitate software development projects as these types of projects are notably knowledge-intensive and require the integration of knowledge from various domains (Patnayakuni, Rai, & Tiwana, 2007; Xia & Lee, 2004). For these projects, the waterfall method brings team members together periodically to elaborately discuss milestone achievements by subject area (Jurison, 1999), while the agile method calls for more frequent yet shorter communication sessions (Cusumano & Smith, 1995; Hass, 2007). However, because these methods set different schedules and frequency for meetings between members, they offer different amounts of time for members to address knowledge shortcomings through meeting or between meeting periods. As such, knowledge acquisition across waterfall and agile methods may create inherent differences in project outcomes of time (schedule), cost (budget), and quality (Drury-Grogan, 2014; Westerveld, 2003). To assess this contention we focus on knowledge acquisition within and across PM methods during meeting and between meeting activities.

We focus on two types of processes that contribute to knowledge acquisition and are generally separated by the temporal nature of PM methods. Specifically, we study the process of team communication that occurs during meeting periods, when members can meet with other members or stakeholders about software requirements. Indeed, researchers indicate that when it comes to project team performance (e.g., Hsu, Shih, Chiang, & Liu, 2012; Park & Lee, 2014), team communication has greater importance than most other Project Management Body of Knowledge (PMBOK) areas (Peltoniemi, Jokinen, & Mönkkönen, 2004). During the project,
team members may communicate with others through formal team meetings, where synchronous communication provides simultaneous interactions, as well as informal exchanges, which utilize asynchronous communication with lagged interactions (Baker, 2002). However, according to media richness theory, the medium restricts the amount of information to be transferred (i.e., verbal, paraverbal, and nonverbal) (Daft & Lengel, 1986), and therefore may affect knowledge acquisition. Thus, we seek to understand the way knowledge acquisition occurs via communication (synchronous and asynchronous) during meeting periods to affect the outcomes achieved across waterfall and agile PM methods.

In between meeting periods, team members are expected to work on their development and delivery of software commitments. During these periods, team members may engage in knowledge acquisition as an independent effort. When working independently, members are likely to acquire knowledge through self-study activities, such as by reading up on a solution approach. Alternatively, members may realize additional knowledge needed from self-study, such as finding a software bug. As such, self-study may lead members to reduce their knowledge gap or realize additional unknowns between meetings. Although the team members' knowledge may be changed independently and at first be tacit and held internally (Nonaka, Takeuchi, & Umemoto, 1996), such knowledge is the basis for knowledge brought explicitly to the team level to drive task success (Nonaka & Von Krogh, 2009). Thus, we seek to understand how knowledge acquisition from self-study between meetings affects team outcomes across waterfall and agile PM methods.

To gain insights and suggest managerial decisions that enhance team outcomes under different PM methods, we employ an agent-based model and conduct virtual experiments. Indeed, researchers suggest that process losses may hinder the team’s ability to fully capitalize on their processes (Kerr & Tindale, 2004; Steiner, 1972), thus limiting the effectiveness of knowledge acquisition processes in providing knowledge. By taking a virtual experiment approach, we examine different probabilities of process losses to communication during meetings or to self-study between meetings to better understand the benefits of managerial decisions that reduce these probabilities. We use a simulation approach because researchers have increasingly employed simulation techniques to provide initial insights into complex communications among organizational members and their tools before testing factors on human subjects (e.g., Chua & Hossain, 2011; Espinosa & Carmel, 2003; Nogueira & Raz, 2006). In particular, agent-based models are beneficial for investigating emergent phenomena that results from communication among agents (Bonabeau, 2002; Davis, Eisenhardt, & Bingham, 2007; Harrison, Lin, Carroll, & Carley, 2007). As such, our approach achieves a targeted investigation of decisions that might positively impact waterfall and agile project teams without any adverse impact to human subjects or organizational effectiveness in the real-world.

Our investigation of waterfall and agile software development projects and the way teams acquire knowledge during meeting and between meeting periods provide at least three contributions to the literature and practice. First, to our knowledge, researchers have yet to provide explicit comparisons of team outcomes across waterfall or agile PM methods. As such, our analysis contributes initial insights about the discrepancy in outcomes by teams using waterfall or agile PM. Second, our consideration of knowledge acquisition process losses from communication and self-study is purposeful in creating propositions about specific managerial decisions. The literature suggests that knowledge sharing is a lever by which organizations make knowledge available (de Vries, van den Hooft, & de Ridder, 2006; Osterloh & Frey, 2000). Our investigation provides insights about knowledge acquisition and the way inefficiencies emerge at different probabilities of process losses. Third, our investigation of knowledge acquisition processes across PM methods examines multiple conditions that affect project time, cost, and quality. We use an agent-based model and virtual experimentation to provide these insights. Researchers advocate that this type of approach provides initial insights into emergent
team phenomenon and should be in the team researcher’s toolkit (Kozlowski & Chao, 2012). As such, our research enriches the study of project methods, dynamic processes, and team outcomes with theoretical and methodological contributions that can move the field forward.

LITERATURE REVIEW AND RESEARCH QUESTIONS

In this section we will provide a high level overview of the background about the two software development methods of waterfall and agile PM methods. Further we highlight the differences in the duration of meeting and between meeting activities between the PM methods.

Project Management Methods for Software Development

The software development process includes the requirements gathering, design, development, testing, deployment, maintenance, and project coordination and management activities for developing a piece of functional software. Not all software development team members will possess the knowledge for each of these activities, which underscores the need for interactions among team members to acquire knowledge (Chau, Maurer, & Melnik, 2003). PM methods serve to provide enough knowledge by having an adequately composed team that interacts around the six PM functions of planning, organizing, executing, monitoring reporting and controlling; and manages the knowledge quality by employing experts (Gannon, 1994). When it comes to software development, which is the set of methods, tools, techniques and models that guide the software design and development process (Hirschheim, Klein, & Lyytinen, 1995), there are many methods that help manage projects. Two methods that have received little direct comparison and are the basis for this research are the waterfall and agile PM methods.

Before we discuss each of these PM methods in more detail, it is worth discussing the project team outcomes that result regardless of PM method. The golden triangle has long been the standard for measuring project outcomes of success, with outcomes defined as the project schedule (time), budget (cost), and quality (Westerveld, 2003). These outcomes measure the success of PM, which is what we are interested in comparing across teams using waterfall and agile PM methods. Externally project success is generally measured against customer objectives; internally, PM success is measured using the golden triangle to help assess the management of a project (De Wit, 1988). These three project outcomes often compete against each another as higher quality generally requires more time and budget to complete (A Guide to the Project Management Body of Knowledge (PMBOK Guide), 2013). Interestingly, research has found that two of the three outcomes, namely schedule (time) and quality, are specifically discussed as iteration objectives in agile teams (Drury-Grogan, 2014). Let us now turn to examine the differences between the waterfall and agile PM methods in more detail.

Waterfall Software Development

The more traditional approach that is considered the foundational standard for software development (Sommerville, 1996) is the waterfall PM method where teams view the optimal software development process as a linear or sequential series of phases for PM (Cusumano & Smith, 1995). This multi-year method began in the 1960s and 1970s for building large-scale software systems that evolved slowly for the U.S. defense and space industries. Waterfall PM was modeled after hardware design projects where engineers could better predict how parts of the system would interact and became known as the classic software development method (Royce, 1970).
Waterfall PM is a unidirectional, top down process with a sequence of activities that is non-iterative (Fitzgerald, 2000; Sommerville, 1996) where phases of the process move from one to the next. In this sequential design process, the progress of a project flows steadily downwards through the phases like a waterfall. One can only move onto the next phase when the preceding phase is completed accurately. According to this method, all requirements are captured before any design and development occurs, meaning the development team rarely interacts with the customer to gain feedback on their understanding of the system requirements. Likewise, each phase produces a specific milestone output with a formal review meeting for that output to uncover any deficiencies, especially for the requirements, design, and coding phases (Cusumano & Smith, 1995).

Agile Software Development

However, in the past waterfall PM was found to be inconsistent in delivering cost-effective and user-driven software to customers (Lyytinen, 1987) as events affecting projects were often unpredictable and projects rarely followed such a sequential flow where customers could identify all requirements upfront (Hass, 2007). As such, a different approach was developed to allow for iterative development of software that would incorporate design changes throughout the project rather than just the end (Fitzgerald, 2000; Sommerville, 1996). The approach, termed agile PM, came about in 2001 to apply a human-centered approach to software development (Fowler & Highsmith, 2001) that delivers high-quality products faster, thereby leading to more satisfied customers (Ceschi, Silitti, Succi, & De Panfilis, 2005). Agile PM allows for simultaneous creation of outputs rather than traditional sequential approach to work flows by waterfall PM (Ballard & Howell, 2003).

Agile PM uses small collaborative software development teams (Dybå & Dingsøyr, 2008) with flexible and adaptable structures where developers are not confined to a particular specialized role but typically self-organize to interchange and blend their roles (Nerur, Mahapatra, & Mangalara, 2005). The project manager’s role as a decision-maker is greatly reduced (Alleman, 2002; Lindstrom & Jeffries, 2004) and the customer or product owner plays a continuous and embedded role in the agile team (Beck, 2000), making them intrinsically involved in many communications on the team. These teams work under extreme time pressure to develop and deliver working software to customers in short iterations (Fitzgerald, Hartnett, & Conboy, 2006; Fowler & Highsmith, 2001). An iteration is a time-boxed period of fixed length, which is often two weeks in duration for agile teams (Schwaber & Beedle, 2002), and iterative development is the process of building a system milestone within a short period of time (Larman, 2004). Because of these short iterations, agile project teams are able to respond quickly to changes in the business, technology and customer requirements by continually redesigning and adapting development processes (Henderson-Sellers & Serour, 2005) as the team only plans for and focuses on one iteration at a time.

Knowledge Acquisition Processes and PM Structure

Project teams tasked with completing milestone outputs may include individuals from different parts of the organization who have different backgrounds and may have opposing ideas and views (Tjosvold, 1987). As such, the individual team members on a software project team may hold expertise in different fields and knowledge domains on the team. We focus on a team member’s knowledge that includes “information possessed in the mind of individuals: it is personalized information (which may or may not be new, unique, useful, or accurate) related to facts, procedures, concepts, interpretations, ideas, observations, and judgments” (Alavi &
Leidner, 2001, p. 109). When members are experts, they have more knowledge and experience in applying that knowledge to problems in a domain, compared to novices, or beginners, who have little experience, are less adept at doing tasks, and who do not have mastery over concepts in that domain (Hinds, Patterson, & Pfeffer, 2001). Thus, we can infer that novices hold more uncertainty about a project and its requirements and need to gather more knowledge than experts in order to complete tasks for that project. Since greater amounts of uncertainty call for more communication (Orlikowski, Yates, Okamura, & Fujimoto, 1995; Yates, Orlikowski, & Okamura, 1999), novices may unintentionally affect processes for knowledge acquisition.

However, the existence of expertise does not ensure success (Denison, Hart, & Kahn, 1996). Rather, researchers suggest that knowledge sharing across expertise boundaries is pinnacle for knowledge creation and, ultimately, team success (Nonaka & Takeuchi, 1995). That is, expert and novice team members must rely on each other for information, group discussion, and decisions to produce a successful output based on the collective contribution of all team members (Katzenbach & Smith, 2005). Further, researchers note that teams rarely achieve optimal outcomes because of process losses (Kerr & Tindale, 2004; Steiner, 1972). Indeed, teams often struggle to share information and collaborate due to the different perspectives pronounced on the team, and other competing priorities and goals, which can result in conflict (Cramton, 2001). They also have difficulty obtaining information due to a lack of participation and interaction of team members (due to the pressure to conform, pressure to present themselves in a favorable way, reliance of junior members on more senior members, domination of an individual, shyness, poor team spirit), sitting in numerous and lengthy meetings (which may not be productive), and miscommunications (Shen, Chung, Li, & Shen, 2004; Wheeler & Valacich, 1996; Whyte, 1993). To better understand and address the situations that enable such negative consequences, we investigate knowledge acquisition process losses from communication during meetings and from self-study between meeting periods.

Knowledge Acquisition from Communication during Meeting Periods

During meeting periods, team members engage in communication to gain knowledge. Specifically, through the process of communication team members pass information (Pinto & Pinto, 1990; Stevens & Campion, 1994; Wang & Ko, 2012) so that members may fill gaps in expertise and coordination (Wang & Ko, 2012). This is important because successful projects depend on members creating, applying, and sharing knowledge (Nonaka, 1994; Nonaka & Takeuchi, 1995; Yuan, Zhang, Chen, Vogel, & Chu, 2009). In order to leverage communication during the project, synchronous communication and asynchronous communication opportunities may be provided to team members. However, the type of communication may limit the amount of knowledge acquired. Media richness theory, advanced by Daft and Lengel (1986), suggests that communication channels differ in cue-carrying capacity (i.e., the types and amount of information that can be effectively transmitted). As such, different communication media lie on a media richness continuum that is anchored by synchronous communication via rich media and asynchronous communication via lean media (Chidambaram & Jones, 1993).

Rich media has a high cue-carrying capacity because it allows for multiple types and amounts of information to be transferred (i.e., verbal, paraverbal, and nonverbal) (Daft & Lengel, 1986). The types of rich media teams use include face-to-face and other tools such as voice over internet protocol (VOIP) or video-conferencing that allow teams to collaborate synchronously. That is, collaboration can occur via information technologies that allow people to interact with each other simultaneously (Baker, 2002; Drury & Williams, 2002). Thus, synchronous communication utilizes rich media that may offer longer exchanges in real time, with more knowledge transferred between seeker and provider.
Asynchronous communication may utilize lean media that is limited in terms of cue-carrying capacity because it restricts non-verbal cues, paraverbal cues, and other types of socio-demographic cues (Straus, 1997). As a result of the limited ability of members to pass certain types of cues or to transmit messages, researchers suggest that teams may exchange shorter messages through these means than through richer media (Boyle, Anderson, & Newlands, 1994). However, lean media often provides a way of sharing information asynchronously through such means as text messengers, email programs, and discussion boards that is essential for meeting across temporal boundaries (Levitt et al., 1994; Ocker & Yaverbaum, 1999). Thus, asynchronous communication via lean media channels produce shorter exchanges, with less knowledge transferred between seeker and provider and does not occur in real-time like synchronous communication. To structure our investigation around meeting activities across PM methods we pose the following research question:

**Research Question 1:** How do process losses to communication (synchronous and asynchronous) during meeting periods affect project outcomes (time, cost, and quality) across PM methods (waterfall and agile)?

**Knowledge Acquisition from Self-Study between Meeting Periods**

Between meetings members may address their knowledge shortcomings through independent self-study approaches. There is limited literature about self-study processes and their impact on team outcomes. As such, we draw on individual based research about learning and team cognition research to understand the ways self-study may address knowledge acquisition. Specifically, through self-study members may find opportunities to learn material, such that the individual gains experience about making the output (Yelle, 1979), and thereby reduces their knowledge need in a particular area. On a software project, members may reduce their knowledge need by reading relevant documentation to better understand the most viable designs. Alternatively, as members apply or acquire knowledge, information processing may lead to realizations that there are inconsistencies and differences with what is currently known by the team (Hinsz, Tindale, & Vollrath, 1997). As such, members may realize additional knowledge needed to complete the team’s task, such as when a software bug or hardware design becomes an impediment to further progress. Although learning and processing information happens at an individual level, members may share knowledge gained or problems realized to enlighten the team (Nonaka & Von Krogh, 2009). Yet, researchers suggest that the knowledge held by individuals may change less rapidly when working alone, than when working as a team (Cooke, Salas, Cannon-Bowers, & Stout, 2000). Thus, self-study between meeting periods is another driver for knowledge acquisition, albeit potentially less effective in the quantity of knowledge acquired than from team activities such as communications during meeting periods. To assess the impact of self-study across PM methods we pose the following research question:

**Research Question 2:** How do process losses to self-study (need reduction and need realized) between meeting periods affect project outcomes (time, cost, and quality) across PM methods (waterfall and agile)?

**METHODS**

To examine and compare process losses across different PM methods, we implement an agent-based model in NetLogo 5.0.3 (Wilensky, 1999) and conduct virtual experiments. Specifically,
we simulate teams completing a project milestone through the acquisition of needed knowledge. The behaviors used to acquire project knowledge include communication (synchronous and asynchronous) during meetings and self-study (need reduction and need realization) that determines outcomes of project time, cost, and quality. Figure 1 describes the steps and algorithmic approaches used in the simulation while the details of the model and experiments are presented below.

**Figure 1. Agent-Based Model Simulation**

**Step 0. Design Parameters**
- Depending on Experiment – Set probability of process losses
- Depending on PM method – Set meeting and between meeting durations

**Step 1. Initialize environment**
- Distribute between meeting opportunities (need reduction, need maintenance, need realization)
- Knowledge Seekers: Initialize knowledge need for 10 Novices, 10 Experts and distribute randomly in environment
- Knowledge Providers: Initialize knowledge owned for 20 Providers and distribute randomly in environment

**Step 2. Start simulation of project milestone**

**Step 3. Meeting Algorithm**
*During meetings*

Move 1 step
- If in contact with a knowledge provider, initialize synchronous communication
  - Acquire knowledge in the amount: provider knowledge owned * P(process loss) * 100%
  - Reduce seeker’s knowledge need
  - Conclude communication
- If no provider available, initialize asynchronous communication
  - Acquire knowledge in the amount: provider knowledge average * P(process loss) * 50%
  - Reduce seeker’s knowledge need
  - Conclude communication
Time advances 1 hour

Perform check for stopping criteria
- If team cumulative knowledge need < 0, stop
- Otherwise continue

**Step 4. Between-Meeting Algorithm**
*Between-meetings*

Move 1 step
- If on need reduction patch
  - Acquire knowledge in amount: provider knowledge average * 20%
  - Reduce seeker’s knowledge need
- If on need realization patch
  - Realize new knowledge need in the amount: provider knowledge average * 10%
  - Increase seeker’s knowledge need
- If on need maintenance patch
  - No change to knowledge need
Time advances 1 hour
Perform check for stopping criteria
- If team cumulative knowledge need < 0, stop
- Otherwise continue

**Step 5. Stopping and Outcome Measures**
When stopped (i.e., team cumulative knowledge need < 0), then record:
- Time
  - Duration, in hours, until stopped (Dur)
- Cost
  - Number of synchronous communications (Sync)
  - Number of asynchronous communications (Asyn)
  - Amount of self-study need reduction (Reduc)
  - Amount of self-study need realized (Real)
- Quality (waste)
  - Maximum amount of knowledge need unmet among seekers (Under)
  - Maximum amount of knowledge need over among seekers (Over)
  - Maximum time, in hours, a seeker had spent in over meeting their knowledge need (Wait)
Model Agents

The simulation environment serves 20 knowledge-seeking agents, representing team members across all project sub-teams (e.g., engineering, programming, sales, and marketing). Past research suggests that small teams with less than four members may have limited knowledge diversity (Jackson, 1996). As such, we expect the sub-teams to include at least four members and be comprised of novices and experts with different levels of domain expertise. However, because we are interested in the lack of project knowledge that drives knowledge-seeking activities, we monitor the gap in domain expertise needed by novice and expert members to complete task goals.

This gap is represented using Poisson random variables with different average values for novices and experts. Based on a general understanding of project milestone iterations, we set the mean values to motivate an expected level of knowledge-seeking during a single project milestone such that novices require at least 5 opportunities to gain knowledge and experts require at least 3 opportunities to gain knowledge. As such, the average novice member’s knowledge need is 40 units (i.e., $X \sim \text{Pois}(40)$) and for expert team members the average is 24 units (i.e., $X \sim \text{Pois}(24)$). This approach creates group heterogeneity similar to the distribution of domain expertise applied to agents by Dionne, Sayama, Hao, and Bush (2010). As well, we assume the mean and variance are equal, hence using the Poisson distribution, thus controlling for overdispersion of knowledge need across team members and more homogenous knowledge need within member types.

Model Activities

We separate knowledge acquisition processes into meeting and between periods that are temporally related to the PM method employed by the team. When waterfall PM is employed, then meeting periods are extended and allotted 8 hours and between-meeting periods are 40 hours. These long time frames provide members with a full day of interaction opportunities that can carry a member through the next week. When agile PM is employed, meeting periods are 4 hours and between-meeting periods are 12 hours. This timing reflects the quick, iterative format of agile (Fitzgerald et al., 2006; Schwaber & Beedle, 2002) such that a half day of meetings is expected to provide enough context for working alone the next day and a half.

During meeting periods members move in the simulation environment along a random trajectory at one step per hour as they seek contact with knowledge providers. There are 20 knowledge provider agents, representing experts or stakeholders with relevant disciplinary knowledge, randomly distributed throughout the simulation environment. The purpose of distributing providers is to mimic the few and far-between synchronous communication opportunities for gaining large amounts of domain relevant knowledge from human sources. The knowledge providers hold a dynamically-created amount of domain expertise generate using Poisson random variables with an average of 8 units (i.e., $X \sim \text{Pois}(8)$).

Members interact with knowledge providers through synchronous or asynchronous communication, each communication requiring one hour in length to account for communication preparation, exchange, and conclusion time. In order to meet synchronously, the member must be in contact with the knowledge provider in the simulation environment. For those members not in contact with a knowledge provider, then asynchronous communication using a lean medium is initiated. Since media richness theory (Daft & Lengel, 1986) indicates that asynchronous communication will provide limited amounts of information, we reduce the amount of knowledge provided to be a maximum of 50% of an average knowledge provider.

During between-meeting periods members continue to move along a random trajectory at one step per hour and come across different self-study opportunities that affect knowledge
need. Members may find opportunities for need reduction of up to 20% of an average knowledge provider, in effect learning project knowledge on their own. Alternatively, members may come across need realization of 10% of an average knowledge provider. Such an effect may occur when faced with a programming bug or a realized uncertainty that needs study. Finally, members may maintain their knowledge, thereby applying what they know until a point when revisions or updates are needed (Smircich, 1983). To distribute the three opportunities, the patches comprising the simulation environment are proportionally divided among the self-study opportunities of need reduction, need realization, and need maintenance.

**Outcome Variables**

The cumulative knowledge need across novices and experts reflects a team-level project knowledge gap (GAP). Although each member is seeking knowledge, we terminate the project milestone when the team-level gap is filled (i.e., GAP < 0, the team-level project knowledge gap falls below zero). Once the team has cumulatively met their knowledge gap, three types of outcomes are measured. First, we assess the cost of work generated by team member behaviors during different periods. For meeting periods we count the number of synchronous communications (SYNC) and the number of asynchronous communications (ASYN) as reflective of communication costs for technology fees, as well as seeker and provider wages. For between-meeting periods, we total the amount of knowledge need reduction (REDUC) and the total amount of knowledge need realized (REAL) that represent the self-study costs for resource access and seeker wages. Second, we record a time outcome as the duration (DUR), in hours, of the project milestone completion. Based on project duration it can be determined how many meeting and between-meeting segments have cycled until completion. Finally, we capture three types of quality measures based on process wastes (Womack & Jones, 2003); these include the maximum amount of knowledge left unmet from under-utilization of opportunities (UNDER), the maximum amount of knowledge that exceeds knowledge need from over-utilization of opportunities (OVER), and the maximum time spent by a member waiting (WAIT), in hours, for the project to complete after the agent’s knowledge need had been met.

**Virtual Experimentation**

To address our research questions about the impact of process losses for knowledge acquisition, we manipulate a probability of process loss from communication during meeting periods (experiment #1) and to knowledge acquisition from self-study between-meeting periods (experiment #2). In each experiment we use a 2 x 5 design (2 PM methods x 5 probabilities of process loss). Specifically, in experiment #1, the probability of process loss from communication (CLoss) is changed from 0.80, 0.60, 0.40, 0.20, and 0.00. In effect, this will prorate how much project knowledge is acquired through synchronous and asynchronous communication so when the probability of process loss is 0.80, P(CLoss = 0.8), then members can only acquire up to 20% of the knowledge offered while 80% is lost. In experiment #2 the probability of process loss from self-study is varied, such that the percent of SLoss assigned to need reduction is changed (SLoss) from 0.10, 0.20, 0.30, 0.40, and 0.50. We hold the amount of probability for need realization at 0.30 or 30% while adjusting the probability for need reduction (SLoss) in the given range. The probability for need maintenance is adjusted with respect to need reduction. For example, when the percent of the SLoss is 0.10, the probability for need reduction is 10%, need realization receives 30%, and need maintenance receives 60%.
RESULTS

We conduct virtual experiments by simulating 1000 project teams in each of the 10 process loss scenarios for a total of 10,000 teams per experiment. To provide comparative insights within each experiment and across experiments, we use the condition with a 40% probability of process loss from communication during meetings and 30% probability of need reduction between meetings, to serve as a baseline, or typical, condition. We begin by comparing the descriptive statistics of the baseline condition across PM methods highlighted in Table 1 and then answer our research questions.

<table>
<thead>
<tr>
<th>Experiment #1: Process Loss from Communication during Meeting Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAP</strong></td>
</tr>
<tr>
<td>0.8</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment #1: Process Loss from Self-study Between-Meeting Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAP</strong></td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experiment #1: Process Loss from Knowledge Acquisition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAP</strong></td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
</tr>
</tbody>
</table>

F-stat 0.80  227.53  403.02  529.84  656.95  784.06  911.17  1038.28

Table 1. Mean Statistics and MANOVA Results

** Note: Superscript values represent Student-Neuman-Keuls (SNK) ranking where samples with the same number have means that are not significantly different. Shaded rows in Experiment #1 represent discussed baseline conditions for waterfall and agile project teams, where P(CLoss = 0.4) and P(SLoss = 0.3); alternatively we could have used the reported statistics from Experiment #2 for waterfall and agile project teams, where P(CLoss = 0.4) and P(SLoss = 0.3); these conditions were equivalent across experiments and led to statistically similar outcomes in all categories.

Baseline Results

In the baseline waterfall condition, project teams have an average project knowledge gap of 639.69 knowledge units (sd = 25.45). Across meeting periods, these teams average 20.68
synchronous communications ($sd = 3.90$) and $338.54$ asynchronous communications ($sd = 110.74$). Between-meetings, on average teams acquire $700.50$ knowledge units ($sd = 95.47$) through need reduction, about $35$ units per member, and increase the knowledge gap an average of $506.02$ knowledge units ($sd = 197.07$), or about $25$ units per member, due to additional need realization. The teams complete the project milestone in $124.70$ hours ($sd = 33.85$) on average, or about $2.6$ meeting and between-meeting cycles in about $16$ days. When the project milestone is complete, there is an average maximum wait time for members with filled knowledge need of $78.24$ hours ($sd = 28.90$); an average maximum knowledge need exceeded from over-utilization of $32.24$ knowledge units ($sd = 8.89$); and an average maximum knowledge unmet from under-utilization of $30.31$ ($sd = 7.78$).

In the baseline agile condition, project teams have an average project knowledge gap of $639.85$ knowledge units ($sd = 25.11$). During meeting periods, agile project teams utilize an average of $27.30$ synchronous communications ($sd = 4.85$) or about $1$ per member. Also, teams utilize, on average, $381.03$ asynchronous communications ($sd = 74.13$), about $19$ per member. Between-meetings, teams acquire an average of $436.34$ knowledge units ($sd = 45.73$) through need reduction, about $22$ units of knowledge per member, but the team gap increases an average of $307.64$ knowledge units ($sd = 94.91$) due to additional need realization, about $15$ units of knowledge per member. The teams overcame their project knowledge gap, and therefore completed the project milestone in $87.29$ hours ($sd = 16.25$) on average; that is about $5.5$ meeting and between-meeting cycles in about $11$ days. At termination, there was an average maximum wait time for members that had filled their knowledge need of $56.27$ hours ($sd = 15.14$), an average maximum knowledge need exceeded from over-utilization of $27.93$ knowledge units ($sd = 6.95$), and an average maximum knowledge unmet from underutilizing of $27.14$ knowledge units ($sd = 6.80$).

**Research Question Results**

To answer our research questions, we assess team outcomes of costs from meeting periods (i.e., total number of synchronous (SYNC) and asynchronous (ASYNC) meetings); cost from between-meeting periods (i.e., total need reduction (REDUC) and need realization (REAL) knowledge units); time as shown by duration (DUR); and, quality (i.e., waste) from waiting (WAIT), under-utilization (UNDER) and over-utilization (OVER). Table 1 shows the means ($M$) for each experimental condition, the MANOVA test results, and Student-Neumann-Keuls (SNK) rankings of significantly different means by experiment. As expected, the test of the team-level project knowledge gap is not significantly different in experiment #1 ($F-stat = 0.71$, $p = 0.70$) or experiment #2 ($F-stat = 0.80$, $p = 0.62$), indicating simulated teams had similar starting conditions. The tests for all other variables indicate that significant differences among means emerged. To organize the results for each research question, we discuss experimental findings across PM methods then compare results to the baseline conditions.

In research question 1, we asked how process losses to communication during meeting periods affect team outcomes across PM methods. Our results, shown under experiment #1 in Table 1, indicate that under the highest level of process loss for communication (i.e., $P(CLoss = 0.8)$) that waterfall project teams have significantly fewer communications during meeting periods on average, with about $1.7$ synchronous and $41.6$ asynchronous communications per member, than agile project teams at about $2.5$ synchronous and $45.8$ asynchronous communication per member (Waterfall $M\{SYNC\|CLoss = 0.8\} = 33.07$ communications, Agile $M\{SYNC\|CLoss = 0.8\} = 50.26$ communications; Waterfall $M\{ASYNC\|CLoss = 0.8\} = 832.82$ communications, Agile $M\{ASYNC\|CLoss = 0.8\} = 916.42$ communications). Because agile project teams need more communication, their meeting behaviors, and therefore costs, are essentially higher than waterfall project teams. The opposite is found for behaviors between meeting
periods where waterfall project teams have significantly more self-study behaviors, need reduction, and need realization on average than agile project teams. While this creates mixed results for costs, the results indicate that waterfall project teams, at about 7.4 weeks on average, take significantly longer than agile teams, that take about 5.7 weeks on average (Waterfall $M_{\text{DUR}\mid \text{CLoss} = 0.8} = 296.53$ hours, Agile $M_{\text{DUR}\mid \text{CLoss} = 0.8} = 229.85$ hours). Further, waterfall project teams have significantly higher, and therefore worse, project quality than agile project teams in terms of members waiting, over-processing information, or under-processing information. Thus, other than accounting for costs, waterfall project teams produce poorer outcomes than agile project teams when there are high process losses from communication.

At the lowest probability of process loss (i.e., $P(\text{CLoss} = 0.0)$), all outcomes are improved, i.e., reduced within PM methods but differences continue to exist across PM methods. While waterfall project teams have significantly fewer communications on average via synchronous communication during meeting periods, with an average of 0.8 per member, than agile project teams at about 1 per member, the number of asynchronous communications is not significantly different (i.e., both at SNK rank 8) (Waterfall $M_{\text{ASYN}\mid \text{CLoss} = 0.0} = 216.53$ units, Agile $M_{\text{ASYN}\mid \text{CLoss} = 0.0} = 244.64$ units). Waterfall project teams have significantly more self-study behaviors, in terms of both need reduction and need realization, between meeting periods on average than agile project teams. Waterfall project teams, at about 2.3 weeks on average, take significantly longer than agile teams, that take about 1.4 weeks on average (Waterfall $M_{\text{DUR}\mid \text{CLoss} = 0.0} = 296.53$ hours, Agile $M_{\text{DUR}\mid \text{CLoss} = 0.0} = 229.85$ hours). While waterfall project teams continue to have significantly higher, and therefore worse, project quality than agile project teams in terms of over-processing and under-processing, the teams are similar in terms of waiting (Waterfall $M_{\text{WAIT}\mid \text{CLoss} = 0.0} = 51.24$ hours, Agile $M_{\text{WAIT}\mid \text{CLoss} = 0.0} = 35.36$ hours). Thus, low process losses from communication continues to create mixed results for costs, although waterfall project teams now align closer to agile project teams based on asynchronous communications for meetings. As well, mixed results for project quality are created such that waterfall and agile project teams have members with their knowledge need filled, waiting for similar lengths of time.

In comparison to the baseline conditions, we see diminishing returns for improving outcomes by reducing process losses from communication. Specifically, incremental reduction in the probability of process loss from communication beyond baseline conditions (i.e., $P(\text{CLoss} = 0.4)$), enables waterfall project teams to make marked decreases in self-study costs from need reduction. However, waterfall project teams only achieve significant improvement to cost from asynchronous communications, time (i.e. duration), and quality (i.e., wait, over-utilization, and under-utilization) when the probability of process loss from communication is reduced to the lowest level (i.e., $P(\text{CLoss} = 0.0)$). Alternatively, agile project teams continue to make significant improvements to a number of outcomes from incremental reductions in the probability of process loss from communication; the benefits are realized in costs from communications (synchronous and asynchronous), self-study costs in terms of need reduction, and duration. Thus, the results indicate that waterfall project teams need greater reductions to the probability of process loss from communication than agile project teams in order to realize improvements beyond the baseline conditions.

In research question 2, we asked how changing the probability of process loss probability from self-study between-meetings affects team outcomes across PM methods. Our results, shown under experiment #2 in Table 1, indicate that at the highest probability of process loss, where the opportunity for need reduction is the lowest (i.e., $P(\text{SLoss} = 0.1)$), waterfall and agile project teams have similar costs from synchronous communications (Waterfall $M_{\text{Sync}\mid \text{SLoss} = 0.1} = 41.18$; Agile $M_{\text{Sync}\mid \text{SLoss} = 0.1} = 40.25$), or about 2 per member. Waterfall project teams have, on average, higher costs from significantly more asynchronous communications and self-study (need reduction and need realization). As well, waterfall project
teams, at about 10.9 weeks on average, take significantly longer than agile teams, that take about 4.0 weeks on average (Waterfall $M(DUR|SLoss = 0.1) = 434.83$ hours, Agile $M(DUR|SLoss = 0.1) = 158.70$ hours). Waterfall project teams have significantly higher, and therefore worse, project quality than agile project teams in terms of members waiting, over-processing, and under-processing. As such, besides costs from synchronous communications, waterfall project teams produce poorer outcomes than agile project teams when there are high process losses from self-study.

At the lowest probability of process loss, where the opportunity for need reduction is the highest (i.e., $P(SLoss = 0.5)$), teams exploit the self-study opportunities for more need reduction, while improving, i.e., reducing, all other outcomes. Waterfall project teams have significantly fewer synchronous and asynchronous communications on average during meeting periods, with an average of 0.7 per member, than agile project teams at about 1.1 per member. Waterfall project teams have, on average, significantly more knowledge need reduction and need realization between meeting periods than agile project teams that likely reflects the greater duration of between meeting duration of self-study. The average duration for waterfall and agile project teams are not significantly different, at about 1.8 to 1.85 weeks on average, respectively (Waterfall $M(DUR|SLoss = 0.5) = 74.18$ hours, Agile $M(DUR|SLoss = 0.5) = 60.17$ hours). Waterfall project teams have significantly higher, and therefore worse, project quality than agile project teams in terms of members over-processing and under-processing. Yet, waterfall and agile project teams are similar in terms of waiting (Waterfall $M(WAIT|SLoss = 0.5) = 45.17$ hours, Agile $M(WAIT|SLoss = 0.5) = 39.00$ hours). Thus, low process losses from self-study creates mixed results for costs, with agile project teams having more communications (synchronous and asynchronous) during meetings, and waterfall project teams having more self-study behaviors (need reduction and need realization). At this process loss level, waterfall and agile project teams achieve similar time outcomes and in terms of quality from time spent waiting.

In comparison to the baseline conditions, we see greater potential for waterfall project teams to improve outcomes from incremental reductions to process losses from self-study. That is, incrementally reducing the probability of process loss from self-study beyond baseline conditions (i.e., $P(SLoss = 0.3)$), enables waterfall project teams to make significant improvements to communication costs (synchronous and asynchronous), and self-study from need realization. Further, waterfall project teams achieve significant improvement to duration and quality (i.e., waiting) from reducing the probability of process loss from self-study by 10% (i.e., $P(SLoss = 0.4)$), that is not significantly different than when the probability of process loss from self-study is changed by 20% (i.e., $P(SLoss = 0.5)$). Agile project teams also make improvements to communication costs (synchronous and asynchronous) with incremental reductions in process loss from self-study beyond baseline conditions. However, the improvements to other outcomes, including self-study costs in terms of need realization, duration, and quality (i.e., waiting) are only realized when the probability of process loss from self-study is reduced to the lowest level (i.e., $P(SLoss = 0.5)$) beyond the baseline condition. Thus, the results indicate that agile project teams need greater reductions to the probability of process loss from self-study than waterfall project teams in order to realize improvements beyond the baseline conditions.

**DISCUSSION**

In this study we focused on improving the dynamic knowledge acquisition processes of team members working on software development projects that are typically coordinated through PM methods, specifically the waterfall and agile PM methods. We considered how members use team communication and member self-study to resolve their knowledge gap in order to effectively contribute to milestones during meeting and between meeting periods. However, we
recognize that the PM method may drive the lengths of these meeting and between meeting periods; thereby affecting the specific managerial decisions needed for adjusting processes to generate desired outcomes. In order to suggest strategies that improve knowledge acquisition processes, we used simulation and virtual experimentation to examine the way knowledge acquisition is affected by two scenarios: (1) process losses from communication during meeting periods; and, (2) process losses from self-study between meeting periods. Herein, we discuss the way our findings suggest specific managerial decisions by PM method and opportunities for future research.

**PM Methods and Baseline Conditions**

We began our analysis with a description of baseline conditions (i.e., $P(\text{CLoss} = 0.4)$ and $P(\text{SLoss} = 0.3)$) for waterfall and agile project teams. In our analysis we use the communication behaviors during meetings and the self-study behaviors between meetings as a proxy for project costs. Interestingly, our results show that waterfall and agile project teams may incur higher levels of costs from different processes. Specifically, under baseline conditions waterfall project teams have fewer communications during meeting periods than agile project teams. Such activity may be representative of agile team interactions where they focus on communication to convey information, even replacing written documentation with informal communication among internal team members and between the team and the customers (Cockburn & Highsmith, 2001). Further, under baseline conditions waterfall project teams engage in more self-study between meetings than agile project teams. This finding aptly reflects the waterfall PM approach of using role-based teams and knowledge sharing primarily through documentation and detailed plans of the entire software development lifecycle (Chau et al., 2003) rather than meetings. Indeed, waterfall PM calls for different individuals or teams to build modules in parallel so that the design, development, and testing takes place by different teams, and these teams only put these modules together and test the entire system in the last phase of the project (Cusumano & Smith, 1995; Jurison, 1999).

While our results suggest that under baseline conditions, waterfall and agile project teams use processes to different amounts during meetings and between meetings, the outcomes that emerge are clearly one-sided. That is, waterfall project teams complete the milestone over longer durations and with more wasted quality. The longer duration may be an artifact of the longer cycles that can propagate inefficiencies, such as process losses. Indeed, research suggests that waterfall excels at control, such that the team uses long periods between meetings to document and capture knowledge gained from the project lifecycle, ensuring the product and process conform to prior plans, support quality improvement initiatives, and satisfy legal regulations (Chau et al., 2003). Such an emphasis on elongating meeting and between meeting periods, however, may create the circumstances for waste. We focus on three types of waste that may emerge due to knowledge acquisition processes. Specifically, members may be left waiting, having fulfilled their knowledge need, essentially completing their project task, and waiting for the rest of the team. These members may also become over-utilized on the project by continuing to collect knowledge and thereby compensate for other members. Further, some members may become under-utilized because they are unable to gather the needed knowledge and cannot fully contribute to the project. In sum, although project teams started with similar team-level knowledge gaps, process losses incurred for more sustained periods by waterfall project teams may have adversely affected outcomes. Thus, we propose:

*Proposition 1: All else being equal for software development, teams using waterfall PM achieve worse time and quality than teams using agile PM.*
Managing Process Losses

In our study, we consider the way different levels of process loss from communication during meetings and from process losses to self-study between meetings affect teams across PM methods. Our results suggest that, in general, the reduction of process losses improves project team outcomes; however, limiting the losses to one type of process or the other enables similar outcomes regardless PM methods. First, we found that reducing the process loss from communication during meetings can lead project teams to incur similar costs in terms of the amount of asynchronous communication across PM methods. The use of asynchronous communication has a limited cue-carrying capacity (Daft & Lengel, 1986). As such, these communications are typically less likely to convey large amounts of knowledge between sender and receiver. While asynchronous technologies of today have created more cost-effective ways to communicate over vast differences, they cannot fully replace the power of synchronous communication (Carmel & Agarwal, 2001). Moreover, when there are process losses from communication, then the knowledge conveyed via asynchronous communication could be further restricted; such losses may come from poorly written messages or other attributes like accuracy, timeliness, coherence, completeness, that diminish the quality of information (Lee, Strong, Kahn, & Wang, 2002). As such, minimizing process losses for communication such that asynchronous communication conveys as much knowledge as possible, while still below that of synchronous, may adequately address the knowledge need under waterfall and agile PM methods. Thus, we propose:

Proposition 2: By minimizing the probability of process losses from communication during meeting periods, teams can achieve similar cost outcomes due to asynchronous communication regardless of PM method.

Second, we found that reducing the process loss from self-study between meetings can lead project teams to incur similar time outcomes across PM methods. That is, by providing team members with more opportunities to reduce their knowledge need on their own, between meetings, teams can complete the milestone more quickly. Interestingly, although researchers note that working in teams can lead to knowledge adjustments more quickly than individuals (Cooke et al., 2000) our findings show that helping individuals engage in self-study is of substantial value to the team. One way of understanding this finding comes from realizing the impact of complexity on projects. On software projects members may be working on innovative designs that are complex; where complexity includes the ambiguity in the requirements of the outcome and multiplicity in the numerous pathways that could be taken to create an outcome (Campbell, 1988). As such, by having the opportunity to study the problem and gain knowledge on their own, the manager is, in essence, allowing members to decompose the complexity of the project and address it in small groups or individually. Moreover, focusing on self-study between meetings and not communications during meetings may be an unexploited strategy to improving team outcomes; indeed, recent research suggests that increasing communication across different communication media can adversely impact team performance (Kennedy, McComb, & Vozdolska, 2011). Given the limited amount of literature about managing self-study processes between meetings in order to promote team functioning and outcomes, we hope that future researchers take up this topic and build upon our initial insights. To direct future inquiries, we make the following proposition:

Proposition 3: By minimizing the probability of process losses from self-study between meeting periods, teams can achieve similar time outcomes regardless of PM method.
Third, by reducing the process loss from either communication during meetings or self-study between meetings, project teams can incur similar quality (waste) outcomes. In particular, reducing process losses and thereby making meetings and between meetings more valuable can reduce the time members spend waiting for others to fill their knowledge need. One way to reduce process losses from communication may be to co-locate team members so that member interactions are rich and valuable. Indeed, collocation allows agile teams to react quickly to rapidly changing or ambiguous requirements by providing ample opportunity for synchronous communication within the agile team, which also reduces the cost of exchange as this synchronous communication is often very precise between team members (Larman, 2004). An approach to reducing self-study losses may come from providing opportunities to prime knowledge individually so that members can activate or apply knowledge when working as a team (Nairne, 1996). This may take the form of assigning individuals or sub-teams to learn about a particular software design approach or bug report and present their knowledge to the team at the next meeting. In effect, taking measures to direct knowledge acquisition processes, can improve the efficiency with which each team member works on the outcomes, and therefore, improves the team-level achievements. Taken together we propose:

Proposition 4: By minimizing the probability of process losses from communication during meeting periods or from self-study between meeting periods, teams can achieve similar quality outcomes in terms of waiting regardless of PM method.

Managing Process Losses to Improve Beyond Baseline Conditions

While our study provided insights about managerial decisions to address different types of team outcomes, the findings also reveal diminishing returns from reducing process losses beyond the level of baseline teams. Indeed, when assessing the effects of reducing process losses from communication during meetings, we find that agile project teams continued to make more improvements from incremental reductions than did waterfall project teams. This further supports the effectiveness of the iterative development by agile teams (Ballard & Howell, 2003; Fitzgerald, 2000) as even incremental reductions in process losses over time is an iterative form of improvement. Rather, waterfall project teams are only able to achieve improvements when large reductions, resulting in minimal process losses to communication, are implemented. As such, project managers attempting to improve outcomes by waterfall project teams, may need to invest extra effort to reduce process losses to extreme levels in order to achieve the desired affects. Reducing process losses from self-study between meetings has the likewise relationship with agile project teams: outcomes are improved beyond that of baseline teams, only when large reductions, resulting in minimal process losses to self-study, are achieved. However, software development team members still need to effectively share domain expertise among team members as not every team member has the knowledge for each of activity (Chau et al., 2003). We capture these arguments with the following propositions:

Proposition 5: Waterfall project teams need greater reductions to the probability of process loss from communication during meeting periods than agile project teams in order to realize outcome improvements beyond the baseline conditions.

Proposition 6: Agile project teams need greater reductions to the probability of process loss from self-study between meeting periods than waterfall project teams in order to realize outcome improvements beyond the baseline conditions.
Limitations and Implications for Future Research

Our investigation assesses knowledge acquisition by agile and waterfall project teams during a project phase using a simulation approach. While we were able to uncover a number of insights about the time, cost, and quality differences among these types of teams, and provide suggestions about mechanisms for improving project outcomes, our study has certain limitations that highlight avenues for future research. First, we made critical assumptions about the behaviors of project team members during meeting and between-meeting activities. Specifically, we assume that team members are purposeful in acquiring knowledge through explicit opportunities in order to complete a task. Researchers suggest that knowledge creation may include socialization, externalization, combination, and internalization (Nonaka & Takeuchi, 1995). As such, our attention to the gaining knowledge is an initial starting point for better understanding knowledge flow on a project team; future research may incorporate other facets of knowledge in order to extend our knowledge-seeking approach.

Second, we assume that team members acquire knowledge from communication with other team members and stakeholders or by working on their own as a self-study. While these constraints helped to focus attention on general behaviors, future studies in other experimental designs, such as in a laboratory or field setting, are needed to provide evidence for our results. Moreover, these generalizations are meant to reflect expected behaviors during PM; however, other, preferential knowledge acquisition behaviors by team members may occur. That is, researchers suggest that team members may preferentially obtain knowledge from social networks sources, internal (Rulke & Galaskiewicz, 2000; Thomas-Hunt, Ogden, & Neale, 2003) and external to the team, (Cummins, 2004). Yet, the integration of management techniques for these types of communication into transitional PM methods is still circumspect. As such, we constrained our focus on the internal communication which may be considered the “known quantities” with which project managers have to work. Thus, there is an opportunity to enhance and build upon the simulation environment we created with knowledge acquisition preferences. However, to adequately capture the dynamics of these preferences, we believe that simulation may continue to be a first step in gaining managerial insights through virtual experimentation before testing mechanisms on human subjects.

Conclusion

Overall, this study has provided both theoretical and practical implications for waterfall and agile teams. The resulting propositions also lay the groundwork for future, continued research with current waterfall and agile PM methods. Specifically, we distinctly compare two often discussed PM methods that are rarely, if ever, compared side by side: namely the waterfall and agile methods. This simulation study provided the opportunity to define 6 propositions than can be used for further study with waterfall and agile teams in practice to understand their communication patterns and resulting effects for process losses and team outcomes. It also provides managerial guidance for waterfall and agile teams in practice. We developed a greater understanding of the knowledge acquisition processes that takes place both during and between meetings. It has long been established that meetings can waste time and accomplish little (Drucker, 1966). However, our propositions provide managerial direction for improving communication during meetings, including asynchronous communication which many teams can benefit from given the global nature of software development. Finally, we specifically examined how communication and self-study can affect project outcomes of project time, cost, and quality. This is challenging as these processes are not as easy to quantify or qualify like hard data metrics such as completion rates, defects, or bugs. Yet knowledge acquisition hinges on individual and team efforts as both are paramount for project team success.
References


