Project Managers' Practical Intelligence and Project Performance in Software Offshore Outsourcing: A Field Study

Nishtha Langer
Indian School of Business, Hyderabad, Andhra Pradesh, 500032, India, nishtha_langer@isb.edu

Sandra A. Slaughter
Ernest Scheller Jr. College of Business, Georgia Institute of Technology, Atlanta, Georgia 30308, sandra.slaughter@scheller.gatech.edu

Tridas Mukhopadhyay
Tepper School of Business, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213; and Carnegie Mellon University in Qatar, Education City, Doha, Qatar, tridas@cmu.edu

This study examines the role of project managers' (PM) practical intelligence (PI) in the performance of software offshore outsourcing projects. Based on the extant literature, we conceptualize PI for PMs as their capability to resolve project related work problems, given their long-range and short-range goals; PI is targeted at resolving unexpected and difficult situations, which often cannot be resolved using established processes and frameworks. We then draw on the information processing literature to argue that software offshore outsourcing projects are prone to severe information constraints that lead to unforeseen critical incidents that must be resolved adequately for the projects to succeed. We posit that PMs can use PI to effectively address and resolve such incidents, and therefore the level of PMs' PI positively affects project performance. We further theorize that project complexity and familiarity contribute to its information constraints and the likelihood of critical incidents in a project, thereby moderating the relationship between PMs' PI and project performance.

To evaluate our hypotheses, we analyze longitudinal data collected in an in-depth field study of a leading software vendor organization in India. Our data include project and personnel level archival data on 530 projects completed by 209 PMs. We employ the critical incidents methodology to assess the PI of the PMs who led these projects. Our findings indicate that PMs' PI has a significant and positive impact on project performance. Further, projects with higher complexity or lower familiarity benefit even more from PMs' PI. Our study extends the literatures on project management and outsourcing by conceptualizing and measuring PMs' PI, by theorizing its relationship with project performance, and by positing how that relationship is moderated by project complexity and familiarity. Our study provides unique empirical evidence of the importance of PMs' PI in software offshore outsourcing projects. Given that PMs with high PI are scarce resources, our findings also have practical implications for the optimal resource allocation and training of PMs in software offshore services companies.

Keywords: IT project management; practical intelligence; software offshore outsourcing

History: Shaila Miranda, Senior Editor; Shaila Miranda, Associate Editor. This paper was received March 21, 2009, and was with the authors 26 months for 5 revisions.

1. Introduction

Software projects often exceed budgets and fail to satisfy end users (Mann 2002). Prior research points to some of the key problems that plague software projects and lead to their failure such as requirements ambiguity, poor change management, lack of quality practices, misalignment of incentives, and stakeholder conflict (e.g., Hirschheim et al. 2002). Offshoring to external vendors increases the cultural, organizational, and geographical distances between clients and vendors (Dibbern et al. 2008, Levina and Vaast 2008) and can create severe information asymmetries that aggravate these problems. For instance, imagine a scenario in which a vendor’s onsite coordinator, a client pleaser, has just agreed to the client’s request to incorporate an out of scope requirement, but without following the existing scope change management process. Integrating this request is detrimental to the project’s costs and the offshore team’s morale, but reneging on this new commitment may jeopardize the client relationship. Flanagan (1954) describes such work related problems

Information asymmetry refers to transactions and/or relationships where one party has more information than the other. This disparity may lead to the use of power and politics in the transactions, which in turn may result in suboptimal decision making (Stigler 1961).
that involve unforeseen and challenging situations as critical incidents.

In software offshore outsourcing projects, the project manager (PM) is the key decision maker who needs the ability to respond effectively to these critical incidents. Prior research on project management has stressed the need for PMs to have skills and experience in communication, leadership, and project-related technology or domain (Napier et al. 2009, Duncan 1996, Thite 1999, Kirsch 2000). However, the dismal rate of success in software projects, reported to be as low as 55% for large projects (Bloch et al. 2012), suggests that although the prescribed skills may be necessary for project success, they are not sufficient. As the preceding example shows, offshore software projects face critical incidents that require nuanced, context-specific management. Traditional project management skills and generic project management processes and frameworks provide broad techniques and guidelines, but cannot readily be tailored to resolve the situation at hand (Odekerken-Schröder et al. 2000). In particular, to resolve the critical incidents inherent in software offshore outsourcing projects, we argue that PMs also need practical intelligence (PI), the PMs’ capability to resolve project related work problems, given their long-range and short-range goals. PI is targeted at resolving unexpected and difficult situations that often cannot be resolved using established processes and frameworks (Wagner and Sternberg 1985, Joseph et al. 2010).

The literature on information technology (IT) outsourcing has focused on contracts and contractual governance as the primary means of addressing problems with information asymmetries in projects (Gopal et al. 2003, Dey et al. 2010, Gopal and Gosain 2010). Unfortunately, these problems cannot be resolved easily with contractual mechanisms: writing a complete contract is not only difficult and costly, it does not accommodate addressing dynamic incidents, whereas writing incomplete contracts that may allow for amendments exposes the clients and the vendors to agency issues and renegotiation costs (Williamson 1979, Hart and Holmstrom 1987). Likewise, the project management literature has prescribed best practices such as processes, tools (Gopal et al. 2002, Ramasubbu et al. 2008), control mechanisms (Kirsch et al. 2002), and risk management frameworks (Wallace et al. 2004). Such best practices, however, are effectively leveraged in more predictable project environments (Morgan 2006). In contrast, as described in the earlier scenario, informational asymmetries and constraints that arise in software offshore outsourcing projects can lead to unexpected, nonroutine, and situational incidents that require contextual responses not readily available through standard operating procedures (Odekerken-Schröder et al. 2000).

Our thesis is that PMs’ PI is salient for project performance in the software offshoring context because project execution spans locational and organizational boundaries (Hirschheim et al. 2002), engendering greater information gaps and a higher probability of unexpected incidents. Logically, the PM, who is ultimately responsible for project success, needs to address the majority, if not all, of these critical incidents. Given that these incidents reframe the project context, do not have a predetermined solution, and cannot benefit from standard processes, the PMs’ ability to detect these incidents, design possible remedies for them, and implement the chosen solutions has a significant bearing on project success (Pich et al. 2002, Simon 1997).

By enabling PMs to detect the nuances in a situation and contextualize responses, PI helps PMs to resolve project-related critical incidents related not only to task, self, and career, but also to numerous inter- and intra-organizational project stakeholders.

Moreover, some project contexts pose informational constraints, rendering PMs’ PI more salient. Prior literature has identified complexity (Brooks 1995, Maxwell et al. 1999, Xia and Lee 2005) and familiarity (e.g., Espinosa et al. 2007, Huckman et al. 2009) as the most pertinent attributes in the project context (the PM, team, or task) that affect information constraints in a project. As we argue in the next section, projects with high complexity or low familiarity are likely to suffer from higher information gaps, increasing the likelihood of more incidents, and thus should benefit even more from PMs’ PI. Given the context-specific management required by such projects (Morgan 2006, Levina and Vaast 2008), it is important to examine how complexity and familiarity moderate the effects of PMs’ PI on project performance.

To evaluate the relationship between PMs’ PI and project performance, we analyze longitudinal data collected in an in-depth field study of a leading software vendor organization in India. Our data include project and personnel level archival data on 530 projects completed by 209 PMs over a four year time period. A key advantage of the data is that we observe multiple projects for each PM; this allows us to isolate and identify the effects of PMs’ PI on project performance. Further, archival data are well suited to studies of performance as they are not prone to response bias or response rates (Boyd et al. 1993, Espinosa et al. 2007). Complementing the archival data, we employ the critical incidents methodology (Wagner and Sternberg 1985, Joseph et al. 2010) to assess the PI of the PMs who led these projects and relate measures of PMs’ PI to project performance. Using rigorous econometric modeling, we find that PMs’ PI has a significant and positive impact on project performance. We also find that projects with higher complexity or lower familiarity benefit even more from PMs’ PI.
Our study makes significant contributions to the literature on project management, which often takes a “universalist” perspective and focuses on best practices to be applied across projects (e.g., Shenhar 2001, Harter et al. 2000, Kirsch et al. 2002, Gopal et al. 2002, Wallace et al. 2004, Project Management Institute 2008, Napier et al. 2009). We extend this literature by (i) introducing and conceptualizing PI as an important capability for PMs that helps in resolving incidents that are situational and embedded in a particular project context; (ii) providing compelling empirical evidence that PMs’ PI is efficacious in improving objective measures of project performance, unlike prior studies of PM capabilities that have been limited to conceptual arguments (Napier et al. 2009) or have found little empirical support (Rai et al. 2009); and (iii) identifying characteristics of the project context—to wit, complexity, and familiarity—that moderate the effect of PMs’ PI on project outcomes (Venkatraman 1989, Shenhar 2001). We also contribute to the literature on offshore outsourcing, which has examined issues such as contracting inefficiencies, but has not often considered PMs’ contribution to project performance. Controlling for contractual, process, and other project attributes, our study highlights the importance of PMs and the effects of their PI on project outcomes in the context of software offshore outsourcing. As PMs with high PI are a scarce resource, our findings also have implications for the optimal allocation of PMs in software services companies, and underscore the need for appropriate nurturing of PI in PMs.

In §2, we present our theoretical arguments. In §3 we describe the field study and data. We present our empirical strategy and analysis in §4, along with the results and a discussion of these results. In §5, we conclude by highlighting the contributions, practical implications, and limitations of our study and identifying opportunities for future research.

2. Theory and Hypotheses

Related Literature

A software outsourcing project involves tasks to be completed, a team dedicated to the project, and a PM who manages the project. Recognizing the key role played by PMs, the software project management literature specifies that PMs should have project management skills (Kirsch 2000), hard skills such as the requisite technical and domain expertise (Thite 1999), and soft skills such as client management, communication, general management, leadership, and team development skills, among others (Napier et al. 2009, El Sabaa 2001). In addition to PM skills, the literature highlights the importance of using standard processes to optimize project delivery (Harter et al. 2000, Gopal et al. 2002), appropriate risk management frameworks (e.g., Wallace et al. 2004), and requisite planning and management tools (Gopal and Gao 2009).

Whereas the project management literature identifies the practices, skills, and tools that optimize project delivery, the information technology (IT) outsourcing literature examines the determinants of outsourcing (e.g., Ang and Straub 1998) and how contracts address information asymmetries and resultant project hazards (Chen and Bharadwaj 2009). In particular, this literature considers the optimal contract choices under various constraints (Gopal et al. 2003, Gopal and Sivaramakrishnan 2008, Kern and Willcocks 2000) and the role of various contract governance mechanisms and contracting structures (Chen and Bharadwaj 2009, Susarla 2012) in ensuring successful outcomes. Prior studies also stress the importance of relational governance (Poppo and Zenger 1998, 2002; Rai et al. 2009), control mechanisms (Kirsch 1997, Kirsch et al. 2002, Gopal and Gosain 2010, Chua et al. 2012), and cultural differences (Levina and Vaast 2008, Ang and Inkpen 2008).

However, despite adequate planning, selecting optimal contracts and appropriate governance mechanisms, and deploying suitable frameworks, processes, and tools, unexpected incidents occur in software projects that cannot be resolved using these mechanisms, leading to project failure (Flyvbjerg and Budzier 2011) and the breakdown of outsourcing relationships (Deloitte 2005). Our study introduces PMs’ PI as an important capability that can be used to resolve such incidents. The PMs’ role within the project requires intangible, tacit knowledge, and contextual management, and PMs’ PI enables PMs to draw upon particular skills when and where needed to respond to the situation at hand. As such, we expand the set of factors that should be considered in understanding how PMs contribute to offshored software projects.

In the next section, we consider projects from an information processing perspective. We draw on the information processing literature to identify how information constraints pertinent to projects can lead to unforeseen incidents or situations where contractual mechanisms or prescribed standardized frameworks are insufficient or ineffective, resulting in adverse project outcomes. We then present PMs’ PI as a capability that is particularly helpful in addressing such problems. Lastly, we hypothesize the moderating effect of the project context on the relationship between PMs’ PI and project outcomes.

An Information Processing Perspective of Software Offshore Outsourcing Projects

To understand how critical incidents arise in a project and why PMs’ PI is a pertinent capability to manage these incidents, we draw upon the information processing literature (Simon 1997, Morgan 2006). Simon (1997) contends that individuals as well as organizations have access only to imperfect information.
Projects, defined as temporary organizational units (Shenhar 2001), also face differing levels of information completeness (Galbraith 1973).

Software projects in general, and offshored software projects in particular, are prone to gaps between information processing needs and capabilities that pose severe management challenges and often lead to unexpected impediments to project success: project stakeholders are distributed across organizations and geographies (Espinosa et al. 2007, Dibbern et al. 2008), PMs routinely interface with clients outside the firm’s boundaries (Hirschheim et al. 2002, Lacity and Willcocks 2001) to deliver complex software, and projects frequently involve unfamiliar technologies and domains (Banerjee and Duflo 2000). To deter the negative effects of incomplete information, the optimal strategy is to reduce the gap between projects’ information processing needs and capacities (Galbraith 1973, Tushman and Nadler 1978). For instance, projects can use standard project management techniques, apply high maturity development capabilities, select appropriate PMs and team members, organize teams to promote information flow, and deploy appropriate task orientation (Kraut and Streeter 1995, Harter et al. 2000, Krishnan 1998).

The literature further suggests that information-processing needs are dynamic, and hence projects can never fully address them. Despite deploying information processing capabilities, there may still be gaps with a project’s information processing needs, or new gaps may emerge as the information processing needs and capabilities change during the course of the project, leading to suboptimal and shortsighted decision making, which in turn sparks unforeseen critical incidents (Simon 1997). We posit that these high levels of information disparities innate in software offshore outsourcing projects increase the likelihood that unpredictable, situational, or unprecedented incidents occur and threaten to derail the project. PMs cannot use standard processes or readily apply a predetermined “cookbook” solution to resolve these critical incidents. Instead, these incidents require the use of organic and contextual management approaches (Morgan 2006, Tatikonda and Rosenthal 2000) such as PMs’ PI. Figure 1 summarizes our research model.

**Figure 1 Research Model**

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Familiarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Software size</td>
<td>• PM task familiarity</td>
</tr>
<tr>
<td>• Team size</td>
<td>• Team task familiarity</td>
</tr>
<tr>
<td>• Schedule</td>
<td>• Project’s stakeholder</td>
</tr>
<tr>
<td>compression</td>
<td>familiarity</td>
</tr>
</tbody>
</table>

PROJECT MANAGERS’ PRACTICAL INTELLIGENCE

H1: +

H2: +

H3: +

PROJECT PERFORMANCE

**PMs’ PI and Project Performance**

Cognitive psychologists (Wagner and Sternberg 1985, Sternberg and Hedlund 2002) refer to PI as a capability that enables an individual to respond appropriately in a particular context, given the individual’s long-range and short-range goals. Often dubbed “street smarts,” PI helps individuals create an ideal fit between themselves and their context through the processes of adaptation, shaping, and selection (Sternberg 1985). PI involves a “response” (behavior) beyond “cognition” for problems that are ill defined or have multiple paths to solutions; it is constrained by both long-term as well as short-term goals; is specific to the actual facts of a particular situation or context; and is manifested in multiple dimensions including task, self, career, and others (Wagner and Sternberg 1985). Although there is considerable debate among cognitive psychologists as to the nature of practical and academic intelligence, PI is argued to supplement academic intelligence (Sternberg and Grigorenko 2001). A typical example of PI at work is knowing which tasks need to be done first and which may be done last in a particular situation, being cognizant of the implications for task completion as well as the impact on others when dynamically adjusting task priority (Joseph et al. 2010). Since PI is contextual, it is difficult to teach in a classroom; more importantly, it is “not acquired automatically with each passing year” (Wagner and Sternberg 1985, p. 450). Table A.4 in the online appendix (available as supplemental material at http://dx.doi.org/10.1287/isre.2014.0523) compares and contrasts PI with academic intelligence and other related concepts.

Drawing on Wagner and Sternberg (1985), we adapt the taxonomy of PI for IT professionals (Joseph et al. 2010) to that for PMs leading software offshore projects and describe PMs’ PI in terms of the ability to manage (i) tasks, i.e., knowing how to accomplish the particular project task at hand; (ii) career, i.e., knowing how to enhance one’s performance as well as reputation by considering long-term career goals; (iii) self, i.e., improving one’s productivity on a day-to-day basis; and (iv) others, i.e., interacting with the various project stakeholders and using social capital (Wagner and Sternberg 1985, Joseph et al. 2010); managing superiors...
to communicate project status; managing peers to cope with the social network within the organization; managing team members (subordinates) for developing, testing, and maintaining developed software; and managing clients to understand idiosyncratic requirements and to facilitate information exchange across firm boundaries (Lacity and Willcocks 2001, Hirschheim et al. 2002, Aron and Singh 2005).

Let us consider the problem of a key resource leaving the project at a critical juncture. This may be a fairly common issue for offshore vendors, but nonetheless requires contextual management. Assume that the key resource is the sole domain expert whose departure at this particular point in the project poses unique and unanticipated challenges, reflected in a sudden elevation of the gap between the project’s information processing needs and capabilities. Given the typical project schedule commitments, the PM cannot fully rely on standard operating procedures to close the gap—for instance, taking legal action against the employee. Thus, she uses her PI to identify the consequences of the key resource leaving and to create innovative solutions by managing tasks, superiors, subordinates, and clients (such as negotiating the timing of the domain expert’s departure, securing an additional team member to be trained by the expert before departing, and assuaging the client’s concerns, if any) to resolve the particular situation.

Although vendors may adopt processes and frameworks that help with routine decision making, the dynamic nature of offshore outsourcing projects may lead to gaps between information processing needs and capabilities, triggering incidents that are rich, complex, and situational, and thus a simple formulaic framework may be inadequate to resolve them. As the primary decision maker, the PM has the ultimate responsibility to ensure that the project runs smoothly. When unexpected and nonroutine incidents occur, it is unlikely that the existing processes or predetermined methods would fully resolve the problem. Whereas extant literature suggests a bevy of skills that successful PMs must possess (e.g., Napier et al. 2009), we posit that PMs use PI to leverage their managerial, interpersonal, and interpersonal skills to craft a context specific solution that best fits the needs of the hour and helps improve project performance (Slaughter et al. 2007).

The nature of PI and the role of the PM in a project suggest that PMs with higher levels of PI can contribute significantly to project performance. The intricate organizational structure typical of offshore software projects may result in misaligned incentives and goals for the vendor and the client, multifaceted reporting structures, and communication glitches across organizational boundaries, exacerbating the resultant information gaps. Consequently, when critical incidents occur, PMs have a view into the task and the various stakeholders associated with the project and thus are in the best position to navigate through the information asymmetries and constraints and the various stakeholder relationships associated with the project to resolve the incident. The PMs’ role within the project requires intangible, tacit knowledge and contextual skills that they can leverage to respond to the situation at hand while managing a complex internal and external organizational structure (Napier et al. 2009). A PM with greater PI would therefore be able to identify critical incidents and resolve them, improving project performance. We therefore postulate the following:

**Hypothesis 1 (H1).** PMs’ PI is positively related to project performance.

**Project Context and PMs’ PI**

Are different levels of PMs’ PI more valuable for different kinds of projects? Software projects vary in their information needs and capabilities and thus may benefit from a tailored management approach. Shenhar (2001) notes that the importance of tailoring project practices to project characteristics has often been ignored in the “universalist” or “best practices” view of project management; thus a more project specific approach is advocated (Andres and Zmud 2002, MacCormack et al. 2003, Slaughter et al. 2006). Differences in project contexts (i.e., in project task, team, and PM) are reflected in differences in the projects’ information processing needs or capabilities leading to information gaps and thus in the likelihood of critical incidents. In particular, the information processing literature suggests that higher project complexity would increase the information needs of a project (Campbell 1988); in contrast, higher familiarity with the task and between stakeholders would increase the information capabilities in a project (Goodman and Leyden 1991, Espinosa et al. 2007). These differences in information constraints in turn affect the extent to which different projects can benefit from higher or lower levels of PMs’ PI, suggesting that PI’s effect on project performance is moderated by the project context. Specifically, we conceptualize moderation as an interaction between PMs’ PI and the level of complexity and familiarity in the project context (Venkatraman 1989). We elucidate on these factors in the following sections.

**Complexity and PMs’ PI**

A complex entity is “one made of a large number of parts that interact in nonsimple ways” (Simon 1981, p. 195). In the context of project management, Baccarini (1996, pp. 201–202) defines complex projects as consisting of many varied interrelated parts, operationalized as “differentiation—the number of varied elements, e.g., tasks, specialists, components; and interdependence or connectivity—the degree of interrelatedness between these elements.” Following Baccarini (1996),
we posit that project complexity consists of (a) technological complexity, defined as the various technological platforms, system requirements, and their scale; and (b) organizational complexity, that is, the multiplicity of project stakeholders, each of which can manifest differentiation and interdependency. Both technological and organizational complexities increase a project’s information processing needs (Wood 1986, Baccarini 1996, Xia and Lee 2005). A more complex project has higher information interdependencies—more pieces to be integrated, and more stakeholders to manage, and also greater information differentiation—diverse technical parts as well as sundry organizational configurations to connect and manage (Campbell 1988, Brooks 1995, Williams 1999, Abdel-Hamid 1990).

Prior literature suggests that when information processing needs increase, context-specific management issues, such as coordination problems with different tasks and stakeholders, tend to arise and hinder project progress (Campbell 1988, Espinosa et al. 2007, Kraut and Streeter 1995). More complex projects impose more severe information processing challenges, increasing the gap between information needs and capabilities and hence the likelihood of critical incidents, thereby requiring greater management effort to achieve project goals (Kraut and Streeter 1995, Mookerjee and Chiang 2002). Further, for such projects, critical incidents may be linked to more interdependent aspects of the project, affect more interdependent tasks, and lead to more severe consequences. Hence, adequately coping with such incidents becomes even more important. We therefore argue that, as project complexity increases, project success is increasingly contingent on PMs’ PI. Whereas ex ante project governance may be adequate for projects low in complexity, ongoing reflexivity is necessary for success as complexity increases. Following Baccarini (1996), we theorize how technological complexity and organizational complexity interact with PMs’ PI to influence project outcomes.

Technological Complexity. Baccarini (1996) describes technological complexity as the difficulties in transforming task inputs into outputs, due to the number of interdependencies or diversity between task elements. In offshore outsourcing, technological complexity is salient as clients are outsourcing larger and more complex software development to offshore vendors with tighter deadlines (Levina and Vaast 2008). Accordingly, in the context of this study, technological complexity is manifested in dimensions that reflect the technical scale and interdependencies in projects—software size and schedule compression.

Software Size. Larger software size contributes to project complexity and consequent management effort nonlinearly (Brooks 1995). When the software to be developed is small, simple, and well defined, task interdependencies can be managed via standard operating procedures and division of labor (March and Simon 1958). Conversely, when the software to be developed is large, task interdependencies are higher, there are more parts that need to be tightly integrated before the software can function as desired, and a substantial amount of information must be exchanged among team members to integrate the pieces. As the software size increases, it becomes more difficult for a single team member to understand which parts of the software and which activities will affect other parts and activities; it also makes it more difficult to prioritize and resolve integration issues (Brooks 1995, Wood 1986, Xia and Lee 2005). Software size is particularly deleterious for offshore outsourcing projects because the requirements analysis and design occurs across firm boundaries (Levina and Vaast 2008). Therefore, a larger software project with its myriad interdependencies is subject to more sources of uncertainties that make it harder to maintain the match between information processing needs and capabilities, making it more prone to critical incidents.

Such critical incidents require nonlinear responses and multiple paths to the solution that cannot be found using formal standard processes and frameworks (Sternberg and Wagner 1993). PMs must draw upon their PI in managing tasks and subordinates to identify the integration issues faced in the large project, identify potential problems that affect one part of the project that may cause problems with other parts, generate alternative solutions, work on quality processes, communicate the tasks and related constraints to the project team, and work with team members to implement them. A PM with higher PI is more likely to understand the myriad relationships between situational variables specific to large projects and find options to respond to critical incidents. Therefore, we expect that larger software projects will benefit even more from having a PM with higher PI and hypothesize the following:

Hypothesis 2A (H2A). PMs’ PI and software size interact positively, such that the positive effect of PMs’ PI on project performance becomes stronger as software size increases.

Schedule compression has been identified as a factor that can considerably contribute to complexity and impact project management (Abdel-Hamid 1990). Offshore vendors have strict contractual agreements that govern timely project completion (Dey et al. 2010), and they have to work under tight schedules. Schedule compression shortens a previously determined schedule in a way that does not reduce and/or minimize the project scope (Project Management Institute 2008). This is typically done by (a) crashing, that is, adding new resources or temporarily transferring project resources...
from other tasks to critical path tasks; or (b) fast tracking, that is, performing tasks in parallel that would normally be done in sequence (Schwalbe 2013). Crashing requires the PM to consider the multiplicity of the task completion information from the additional team members; fast tracking necessitates that the PM manage multiple distinct task paths and schedule them in tandem (Schwalbe 2013). Both techniques add to the complexity of managing the project by increasing not only the technological interdependencies, but also the differentiation within the project. The rate at which the PM must process information, integrate information inputs, integrate project resources, and make management decisions contributes to project complexity, thus altering the technological steady state factors. When schedule compression is high, the PM must intensely manage a greater number of events for the work to be accomplished within a tight timeframe (Mookerjee and Chiang 2002). Schedule compression levies information processing constraints that existing information processing capabilities may not address, leading to a higher likelihood of critical incidents. In such situations, PMs with higher PI would be able to better manage tasks and various stakeholders and make dynamic management decisions. These PMs would readily spot the paths that may slow down progress and shift resources accordingly or reprioritize project goals or reschedule tasks so that all paths can be completed simultaneously, while effectively managing the impacts of the changing dynamics in the project schedule on the different project stakeholders. Therefore, we have the following:

**Hypothesis 2B (H2B). PMs’ PI and schedule compression interact positively, such that the positive effect of PMs’ PI on project performance becomes stronger as schedule compression increases.**

**Organizational Complexity.** Organizational complexity in projects involves situations where the stakeholders are numerous and diverse, and it involves a dynamic interdependent organizational structure (Baccarini 1996), as is often the case with offshore outsourcing (Carmel 1999). In the context of this study, organizational complexity is manifested in team size and team dispersion, as we discuss.

**Team Size.** Given the increasing complexity of offshored software projects, vendors often deploy large teams to achieve project goals (Levina and Vaast 2008). However, larger teams also exponentially increase the number of interdependencies and coordination links among team members, contributing to the project’s organizational complexity (Baccarini 1996, Xia and Lee 2005). Larger teams add to complexity in two ways. First, a PM needs to allocate team members optimally. Since this allocation can entail different permutations and combinations, there are a large number of desired outcomes that need to consider the interdependencies between different team members and the diversity of roles that each plays within the team, requiring more information processing. Second, larger teams are more difficult to manage because there are more people to supervise. A team of n members can have as many as n(n−1)/2 dyadic links and additional higher degree links, which increase project management overhead (Brooks 1995). Thus, a larger team size increases the project’s information flows, imposing more severe information constraints that increase the likelihood of critical incidents occurring. A PM can leverage PI to resolve these incidents and manage subordinates by identifying (a) communication and coordination difficulties, (b) alternative solutions that fit the situation, and (c) implementing them, improving project outcomes. For example, the PM could resolve specific coordination problems by identifying the most favorable team allocation (Faraj and Sproull 2000) in a specific context or by adapting communication practices to what is needed for the project (Napier et al. 2009). Higher PI can help PMs to better manage the incidents that occur in projects with larger team sizes; we thus propose the following:

**Hypothesis 2C (H2C). PMs’ PI and team size interact positively, such that the positive effect of PMs’ PI on project performance becomes stronger as team size increases.**

**Team Dispersion.** In most offshore projects, team members and stakeholders are dispersed across organizations and/or in different locations. Such projects usually involve an onsite client coordinator, onsite team members who gather requirements, and an offshore team that implements or codes the software. The geographic dispersion needed to deliver offshore development projects manifests in inherently complex organizational structures. The onsite team has a greater understanding of the client’s needs, whereas the offshore team has a greater understanding of delivery and software development processes (Rao and Hoyt 2007, Levina and Vaast 2008). Both teams have to execute interdependent tasks in sync to deliver a project successfully (Carmel 1999). In addition, recall that the onsite and offshore teams lack a shared context and have dissimilar roles, diverse incentives, myriad reporting structures, distinct languages, and a different work culture, all of which increase the teams’ differentiation (Hinds and Mortensen 2005, Levina and Vaast 2008). Greater team dispersion increases the coordination and incentive challenges between geographically dispersed software development teams (Herbsleb and Mockus 2003, Espinosa et al. 2007). The opacity in project goals and the difficulty of resolving team conflicts in dispersed teams increase a project’s information constraints and hence the likelihood of critical incidents.
Unlike colocated teams, geographically dispersed teams cannot rely upon face-to-face communication that meets their information needs and helps resolve these incidents. These teams must frequently overcome barriers to communicating such as different languages, cultures, and time zones (Carmel 1999, Orlikowski 2002, Espinosa et al. 2003), which contribute to coordination challenges. Such teams fail to communicate and retain contextual information or to determine the salience of information, creating severe impediments for mutual knowledge that is essential in project completion from teams (Cramton 2001). The example discussed in the introduction described a situation where the onsite coordinator commits to a new requirement without understanding the impact of such an action on the offshore team’s ability to deliver. A PM with high PI will go beyond standard communication protocols and proceed to manage the two offending stakeholders: in this case, the offsite coordinator (a subordinate) and the client. Rather than just proving that the stakeholders are wrong and potentially endangering the project’s success (which could also have implications for the PM’s career), the PM should draw on PI in managing task, self, career, and others and proceed by explaining the impact of the new requirement on the project and by initiating a renegotiation on project schedule and cost agreements, if necessary. In sum, projects with a higher degree of team dispersion have greater information gaps and are likely to have more critical incidents. Therefore, they will benefit even more from PMs endowed with high PI. Hence, we have the following:

Hypothesis 2D (H2D). PMs’ PI and team dispersion interact positively, such that the positive effect of PMs’ PI on project performance becomes stronger as team dispersion increases.

Familiarity and PMs’ PI

Goodman and Leyden (1991) define familiarity as the specific knowledge people have about their work—the requisite tools, work environment, team, and performance strategies. Familiarity thus refers to specific knowledge about the unique aspects of the work environment—the task, setting, or crew (Goodman and Leyden 1991). Familiarity has been identified as particularly pertinent to offshore outsourcing (Huckman et al. 2009). In the software offshore outsourcing context, familiarity is reflected in task familiarity and stakeholder familiarity, that is, prior cognizance of the task or project stakeholders such as the team or the client (Krishnan 1998, Espinosa et al. 2007, Huckman et al. 2009). Task and stakeholder familiarity not only affect project outcomes, but by changing the information constraints in the project, also influence the effect of PMs’ PI on project outcomes. From the information processing perspective, low familiarity projects have significant information processing constraints because the task, team, or client are unfamiliar to the PM or the team members are not familiar with the task or with each other. Hence, projects with low levels of familiarity are more prone to communication gaps, inter-personal or inter-organizational conflict, and glitches in task execution (Barki et al. 1993).

For example, consider a hypothetical situation where the vendor in question has assigned freshly hired graduates to the team. Thus, the majority of the team members are unfamiliar with each other and unable to coordinate implicitly with each other (Crowston and Kammerer 1998) or to locate expertise within the team to help them execute their task (Faraj and Sproull 2000). It is expected that such a project would generate many critical incidents for the PM because the project’s information processing needs to coordinate teamwork are likely to inundate its information processing capabilities, increasing the information gap. A PM with greater PI would be more useful in situations characterized by such an imbalance as she will be able to dynamically reconfigure responsibilities for team members and redefine the coordination structure within the team. We now discuss each facet of familiarity.

Task Familiarity

Generally, task familiarity refers to the knowledge that individuals have about the aspects of their tasks (Goodman and Garber 1988, Espinosa et al. 2007), which in the context of software outsourcing refers to prior technology, domain, and methodology expertise relative to what is required for the current project (Krishnan 1998). First, technological expertise appropriate to the software project is critical to its execution. Second, domain expertise is crucial when interacting with the client and the design team because the focus of software activities centers on the effective application of technology to meet business needs (Lee et al. 1995). Third, the literature suggests that maintenance and development projects require different methodological approaches (Swanson and Beath 1990). Task familiarity is relevant for the PM as well as the project team. The PM is likely the senior most member of the project team and is often perceived as a sounding board for technical and architectural decisions (Lee et al. 1995). Since software development and maintenance require extensive management of task-related expertise, task familiarity with project technology, domain, or methodology benefits a PM in such tasks. Task familiarity at the team level also similarly helps task execution.

Although task familiarity for both the PM and the project team is desirable, it is not always possible to obtain a team and PM allocation that harmonizes with the project’s task attributes. When a team or a PM is unfamiliar with the task at hand, the information gaps in the project increase significantly because it
is not clear what must be done and how to do it. Thus, there is a higher likelihood that the software produced may have quality issues, require rework, cannot meet performance objectives, and have other similar incidents (Kirsch 2000, Espinosa et al. 2007). These critical incidents have to be actively resolved so as to minimize their negative impacts on project outcomes.

We propose that projects with low task familiarity can benefit more from PMs with higher PI. When critical incidents arise because of information constraints in projects with low task familiarity, PMs must draw upon their PI to manage task, self, and others by readily communicating with project stakeholders and acquiring the appropriate technical know-how needed to effectively resolve the particular incidents. Specifically, PMs can use their interpersonal and managerial skills as needed by the situation to work with other team members (Napier et al. 2009) to discover the important elements of the task that need to be coordinated and managed in the project, or even put immediate plans to enhance such expertise as the project proceeds. Using PI, PMs can analyze specific problem contexts and tap the requisite expertise within the team if possible, or locate that expertise elsewhere with the help of superiors or peers as and when information gaps increase and critical incidents arise (Faraj and Sproull 2000). The PMs may use PI to devise ways to compensate for a deficiency in expertise, by dynamically rearranging the tasks across team members when possible, or by acquiring temporary access to such expertise outside the team as needed to resolve a particular incident. Thus, we have the following:

Hypothesis 3A (H3A). PMs’ PI and PMs’ task familiarity interact negatively, such that the positive effect of PMs’ PI on project performance becomes stronger as PMs’ task familiarity decreases.

Hypothesis 3B (H3B). PMs’ PI and team’s task familiarity interact negatively, such that the positive effect of PMs’ PI on project performance becomes stronger as team’s task familiarity decreases.

Stakeholder Familiarity
Goodman and Leyden (1991) describe how familiarity with the setting and the crew improves productivity at work. In the context of software outsourcing, client familiarity is relevant as the setting, and crew familiarity is easily translated to team familiarity; the two constitute stakeholder familiarity (Espinosa et al. 2007, Huckman et al. 2009). Stakeholder familiarity is very relevant for software offshore outsourcing projects (Aron and Singh 2005, Levin and Vaast 2008); prior literature has suggested the use of appropriate software project management practices to manage a project’s stakeholders (MacCormack et al. 2003). However, low stakeholder familiarity increases information processing constraints and results in the likelihood of more critical incidents that may not be resolved using such practices: the team members may not have access to shared knowledge or an expertise pool, or an unfamiliar client setting and expectations may lead to misunderstandings about the project objectives (Reagans et al. 2005, Aron and Singh 2005). Low team familiarity prevents team members from interacting and coordinating implicitly, increases the information gaps and asymmetries in the project, and requires greater management effort (Crowston and Kammerer 1998, Faraj and Sproull 2000, Espinosa et al. 2007). Similarly, low client familiarity precludes PMs from knowing how the client operates, and by placing PMs in an unfamiliar context, increases the project’s information requirements.

PMs need to draw on PI to devise a number of alternative solutions to incidents arising because of greater information imparities that result from low stakeholder familiarity. For example, for unfamiliar teams, PMs need to use PI to determine the deficiencies present in specific situations and find appropriate remedies by (i) locating requisite expertise within the team or the broader organization and making the expertise immediately accessible to appropriate team members who are in need of help, (ii) persuading unfamiliar team members into sharing expertise to help solve the problem at hand, (iii) helping team members collaborate in specific instances, and (iv) encouraging joint efforts over individual contributions if the situation so demands. Similarly, when the client is new and unfamiliar, PMs use PI to resolve the critical incidents that arise because of this information gap by considering a wider range of possible client reactions, paying particular attention to initial client interactions, and being aware of client needs and procedures that may require unusual solutions. For a particular problem, PMs should draw upon PI to facilitate cross-boundary interaction and joint problem solving (Rai et al. 2009). On the other hand, in a project in which stakeholder familiarity is high, there may be a greater shared understanding of what the project is supposed to accomplish and stakeholders can rely upon their familiarity to interact and coordinate more effectively. Stakeholder familiarity will reduce the information gaps and decrease the likelihood of critical incidents that require intervention and resolution by PMs. Therefore, we propose the following:

Hypothesis 3C (H3C). PMs’ PI and project’s stakeholder familiarity interact negatively, such that the positive effect of PMs’ PI on project performance becomes stronger as stakeholder familiarity decreases.

3. Methodology
Research Setting
To empirically validate our hypotheses, we conducted an in-depth field study at a leading software out-
sourcing vendor in India. The vendor provides an ideal setting for our study. The company is very large, employs thousands of project personnel, and completes many diverse software projects each year. The vendor also deploys stringent quality, risk management, and project management processes and frameworks and has been assessed at capability maturity model (CMM) level 5; as a CMM level 5 organization, the company collects numerous measures of projects, project personnel, and performance. The company was at CMM level 5 during the entire time period of our study. In a typical scenario at the company, when commencing a new software project, a senior manager selects a PM out of a pool of available PMs to lead the project. The selection of PMs for projects can be constrained by their availability as well as by other typical constraints such as their travel and visa status. Our discussions with senior managers revealed that although they would like to try to pick a PM that fits the anticipated project profile, they did not measure PMs' PI, and thus used whoever was available to get the job done.

Measuring PMs’ PI

We use the critical incidents approach to assess PI in software project managers (Wagner and Sternberg 1985, Hedlund et al. 2002, Joseph et al. 2010). Specifically, we gathered critical incident scenarios from nine project management experts at the research site. Table A.1 in the online appendix provides the number of critical incidents collected for each of the seven dimensions of PI. Following is an example of a critical incident provided by the experts: “You are the offshore PM for a project. The project faced some delays in the timelines, eventually cutting into the time for testing the product thoroughly. On one hand, you need to make an on-time delivery as promised to the client. On the other hand, you are sure that the deliverable is not of the quality it should be. Describe what actions you would take.” Table A.5 in the online appendix provides additional examples of the critical incidents provided by the experts.

Next, we developed a web-based instrument to administer the incidents to 300 randomly selected PMs at the company. Each PM responded to the questions by writing a detailed essay type of response for each incident. We received complete responses from 209 PMs and conducted statistical tests to check for nonresponse bias. T-tests of means for experience and performance ratings for the PMs who completed the PI instrument and the 300 PMs (in the original data set) indicated no significant differences, suggesting no bias in terms of response or nonresponse to the critical incidents.

An expert panel of four evaluators, each with graduate degrees and considerable experience in IT and management of software outsourcing projects, evaluated these responses (this is consistent with the approach used in Hedlund et al. 2002 and Joseph et al. 2010). The experts evaluated each response on a scale of 1–7, ranging from 1 (extremely poor response) to 4 (average response) to 7 (extremely good response). We obtained 5,852 evaluations (i.e., 4 × 209 × 7) in all. We computed the $r_{xx}$ measure of agreement between raters (James et al. 1984) for the evaluations of each aspect of PI, and found that the measures of agreement all exceeded 0.95 (see Table A.2 in the online appendix). Given the high level of agreement, we averaged the ratings of the four raters, yielding average measures of each of the seven aspects of PI for each of the 209 PMs. Finally, to assess the dimensionality of the PI evaluations, we conducted an exploratory factor (principal components) analysis with the eigenvalue > 1 criterion (Harman 1967). The input to this analysis was the average ratings for each of the seven aspects of PI for each PM, that is, task, self, career, superiors, peers, subordinates, and clients. This analysis suggests the presence of a single factor among the ratings. This factor explained about 40% of the variation in the ratings data, and loadings of the evaluations on the factor ranged from 0.40 to 0.70. Given these results, we used the factor score from the one factor model as a measure of PMs’ PI (PI). The online appendix provides further details about our measure of PI and the processes we used to evaluate its validity.

Archival Project Data

We collected archival data on 530 software outsourcing projects executed between 2002 and 2006 that were managed by the 209 PMs for whom we had the PI measure. These data include detailed project-level financial, resource allocation, project characteristics, client feedback, and personnel data. We were able to collect data on multiple projects per PM. This is critical for our analysis as, without multiple projects per PM, it is not possible to statistically identify the effects of PMs’ PI on project performance.

2 Subsequent analyses support this assertion as regressions relating project characteristics including measures of complexity (software size, schedule compression, team size, and team dispersion) and measures of familiarity (PM’s task familiarity, team’s task familiarity, team familiarity, and PM’s client familiarity) to PMs’ PI yield insignificant coefficients, suggesting that there was no distinct pattern of matching PMs to project characteristics at the vendor organization over the time period of our study.

3 At least two projects are needed for a PM in order to distinguish the effects of the PM on project outcomes. Collecting multiple projects per PM constitutes a panel data design in which multiple observations are collected for a PM over time. This design provides an opportunity for the researcher to make statements about the cause-effect relationships among different variables (which is not possible with a cross-sectional design). For more on this identification strategy, see Hsiao (1986) or Banerjee and Duflo (2000).
Measuring Project Outcomes

Software projects have multiple outcomes. Our study focuses on cost performance and client satisfaction (Rai et al. 2009). Cost performance is measured as the actual costs incurred by the project. Since an increase in costs, relative to the size of the project, is akin to a decrease in performance, we reverse score our measure of cost performance by multiplying it by a $-1$. Further, the cost performance data were skewed, i.e., not normally distributed; to correct for this, we use a log transformation (Greene 2002). Therefore, we follow Espinosa et al. (2007) and use $(-1 \cdot \log(Cost))$ as our cost performance measure ($CostPerformance$). In the context of outsourcing, client satisfaction is a critical dimension of project performance (e.g., Aladwani 2002). The client satisfaction rating is the result of a meticulous process that involves periodic project review meetings with client representatives who are intimately involved in the project. We measure client satisfaction ($ClientSatisfaction$) with a single item of the actual client rating of satisfaction for each project. The rating was numeric on a scale of $1$ (very dissatisfied) to $7$ (very satisfied), but not constrained to whole numbers. For example, a typical rating could be $5.5$.

Measuring Complexity and Familiarity

As discussed earlier, complexity is assessed using software size, schedule compression, team size, and team dispersion. In particular, software size is measured in terms of function points (Kemerer 1993); our measure of software size uses the log transformation of the estimated function points for the project ($logFP$). Schedule compression ($ScheduleCompression$) is measured as the ratio of the optimal project duration as recommended by the constructive cost model (COCOMO) (Boehm et al. 2000) to the actual project duration derived from archival data (Abdel-Hamid 1990). Team size ($TeamSize$) is computed by counting the total number of individuals assigned to work on the project (Shenhar 2001). The team’s geographic dispersion ($TeamDispersion$) is computed as the ratio of the onsite versus offshore work, as measured in person hours (Espinosa et al. 2007).

Likewise, familiarity is measured using task and stakeholder familiarity, assessed for both the PM and the team. We conceptualize task familiarity as an individual’s familiarity with the specific technology, domain, and methodology relevant to the project at hand (Krishnan 1998). For example, if the current project is a development project in the telecommunications domain using Java, and the project allocation data indicate that the PM has four years of experience in development, three years in telecommunications, and two years using Java, we can calculate a PM’s task familiarity ($PMTaskFamiliarity$) for this project as $4 + 3 + 2)/3 = 3$. To measure team task familiarity ($TeamTaskFamiliarity$), we sum the computed measure for the team and normalize it by the team size.

Stakeholder Familiarity

The literature on familiarity suggests that stakeholders develop familiarity based on the work they complete together, rather than, for example, social interactions or social relationships (Goodman and Leyden 1991). By completing work together, the stakeholders develop a common knowledge base upon which they can draw (Espinosa et al. 2007). Thus, familiarity is typically measured in terms of the number of work units or number of years of working together. To measure team familiarity, we assign a count variable for each team member, and increase the count variable by one if a team member had worked with another member prior to the current project (Reagans et al. 2005, Espinosa et al. 2007, Huckman et al. 2009). We repeat this for each team member, sum it up at the project level, and normalize it by the number of team dyads ($TeamFamiliarity$). To measure PM’s client familiarity ($PMClientFamiliarity$), we count the prior years of experience of the PM with the current project client.

Interaction Variables.

We operationalize eight interaction variables in which we multiply each of the complexity and familiarity variables by our measure of PM’s PI (Venkatraman 1989). Further, we standardize each variable prior to analysis to facilitate interpretation of coefficients and to mitigate collinearity issues in estimation (Aiken and West 1991).

Controls.

While we posit PI as a capability for ad hoc governance of the project, projects would use ex
ante governance mechanisms at the project level, the PM level, and the firm level. Thus, our controls include project level controls (contract type, technology, project type), PM level controls (PM work experience, PM rating), and firm level controls (firm level learning, and vendor’s client familiarity), as indicated by the prior literature.

**Project Level Controls.** Gopal et al. (2003) find that different project contracting types can lead to variation in project outcomes. We therefore include a binary variable (ContractType) to distinguish between time and materials contracts (= 1) or fixed price contracts (= 0). In addition to contract type, we control for the project’s programming language, as different programming languages have different productivity levels (Jones 1996). We use a binary variable (ProjectPLP) to control for such productivity differences based on the project’s programming language, whether low productivity (= 1) or high productivity (= 0). Similarly, we include a binary variable (ProjectType) to distinguish between maintenance (= 1) and development (= 0) projects, because these may use different management styles and processes (Swanson and Beath 1990).

**PM Level Controls.** Prior experience as a PM influences how an individual manages a project and influences outcomes (Kirsch 2000, Joseph et al. 2010). We therefore control for the PM’s prior project management experience (PMWorkEx) at the start of a project. We also control for PM’s past performance by using a measure of the PM’s past performance ratings (PMRating) wherein we average historical performance ratings data for each PM at the start of the project.

**Firm Level Controls.** We control for firm learning effects by including a measure of the number of prior projects completed at the firm level (PriorProjectsOrg), prior to the start of the project. We also include a dummy variable to indicate the vendor firm’s client familiarity (dVendorClient), whether the client was familiar to the vendor (= 1) before the current project execution.

Table 1 presents the descriptive statistics and the correlation matrix for the model variables.

### Analysis, Results, and Discussion

**Analysis**

As we collected data on multiple projects for each PM, and as each PM completes one project at a time, our data form a panel that is dimensioned by PM \(i\) and project \(j\). To test our hypotheses, we developed the following system of equations for cost performance and client satisfaction:

\[
\text{CostPerformance}_j = \alpha_0 + \alpha_1 \cdot \log(FP)_j + \alpha_2 \cdot \text{ScheduleCompression}_j + \alpha_3 \cdot \text{TeamSize}_j
\]

![Table 1: Descriptive Statistics and Correlation Matrix](image)
and models met the normality assumptions of ordinary least squares (OLS) and SUR. We also computed condition indices (Belsley et al. 1980) and variation inflation factors (Marquardt 1970); the highest VIF computed was 3.69. We further tested for heteroskedasticity using White’s test (White 1980), and for autocorrelation using the Durbin-Watson test (Greene 2002), and did not find any problems. As there are multiple projects per PM, the SUR estimation for panel data uses random effects to correct for any potential correlation of residuals across projects that are nested within the PM (Greene 2002, Biørn 2004). In our SUR estimation, we added the dummy variable dVendorClient, indicating the vendor’s prior client experience, to the client satisfaction equation, to render the parameters in the two equations different and identify the system of equations (Wooldridge 2002). When the vendor knows the client, it may better understand the client’s domain and the nature of the work to be done. This knowledge can help the vendor design better products and/or provide better service for the client, leading to higher client satisfaction. However, prior client experience (at the vendor level) may not directly influence project costs, which are more a function of the specific project’s requirements (this is borne out empirically as the correlation between dVendorClient and CostPerformance is −0.01 and insignificant).

Results
Following common practice when analyzing models with interaction effects (Cohen and Cohen 1975, Aiken and West 1991), we estimated our models hierarchically. We first tested the baseline models for cost performance and client satisfaction that include only the control variables. Next, we added the PI variable to the baseline models with the control variables. This increased the explanatory power of the models significantly, especially for the client satisfaction model (CostPerformance: $\Delta R^2 = 0.05$, $F_{4,692} = 22.249$, $p < 0.001$; ClientSatisfaction: $\Delta R^2 = 0.11$, $F_{4,692} = 70.50$, $p < 0.001$), suggesting that PMs’ PI explains significant incremental variation in both cost performance and client satisfaction. We then added the complexity interactions to the PI model, and found that the change in $R^2$ is significant, implying the significance of the complexity interactions (CostPerformance: $\Delta R^2 = 0.03$, $F_{4,692} = 8.11$, $p < 0.001$; ClientSatisfaction: $\Delta R^2 = 0.04$, $F_{4,692} = 6.75$, $p < 0.001$). Similarly, we added the familiarity interactions to the PI model, and again found evidence of the significance of the familiarity interactions (CostPerformance: $\Delta R^2 = 0.03$, $F_{4,692} = 9.28$, $p < 0.001$; ClientSatisfaction: $\Delta R^2 = 0.04$, $F_{4,692} = 6.04$, $p < 0.001$). Finally, we esti-

Panel data analysis could include fixed or random effects. However, in our panel data set, PI is time invariant and would drop out when fixed effects are used to estimate the equation; thus we use a random effects estimation in our analysis. Specifically, we estimate our model using the xtsr specification in Stata; xtsr fits a many-equation SUR model for panel data using random effects estimators (Biørn 2004).
imated the full model by adding all of the interaction variables for complexity and familiarity. We again found that the predictive power of the model increased (\(\Delta R^2 = 0.062, F_{4,42} = 9.999, p < 0.001\); \(\Delta R^2 = 0.126, F_{6,42} = 11.686, p < 0.001\)), thus the interaction variables explain significant incremental variation in both outcome variables over the baseline and PL models. The condition indices for the SUR specification were 12.090 (baseline model), 13.406 (+PI), and 13.317 (+interactions). Our results are shown in Table 2.

**PI and Project Outcomes.** We find support for H1; the coefficient for PI is positive and significant for both cost performance and client satisfaction (\(\alpha_6 = 0.44, p < 0.01; \beta_6 = 0.644, p < 0.01\)).\(^{11}\) A higher level of PMs’ PI is thus significantly associated with both higher cost performance and client satisfaction.

---

\(^{11}\) Note that the main effect of PMs’ PI is tested by differentiating the full model with respect to PI (Greene 2002). Because we use standardized (zero-mean) variables for complexity and familiarity in our analyses, all terms drop out of this derivation except for the main variable PI, and we report its coefficient and significance in testing H1.

**PI, Complexity, and Project Outcomes.** Hypothesis 2 posited that PMs’ PI would be more helpful for projects that are more complex. We find support for H2 for different complexity variables. As reported earlier, the change in \(R^2\) squared when adding the complexity interactions to the baseline model is significant at the \(p < 0.001\) levels in both the cost performance and client satisfaction models. Individually, the interactions for software size (\(z = 2.87, p < 0.01\)), schedule compression (\(z = 10.60, p < 0.01\)), and team size (\(z = 2.26, p < 0.05\)) are significant in the cost performance model, whereas the interaction for team dispersion is insignificant in this model (\(z = 1.28, p > 0.10\)). In the client satisfaction model all of the individual interactions for software size, schedule compression, team size, and team dispersion are significant, although the interaction effect for team dispersion is in the opposite direction to that hypothesized (\(z = 4.75, p < 0.01\); \(z = 6.30, p < 0.01\); \(z = 3.15, p < 0.01\); and \(z = -3.49, p < 0.01\), respectively).

**PI, Familiarity, and Project Outcomes.** Our findings suggest that project familiarity—for task and stakeholder—is likely to be substitutive with PMs’ PI, as posited by H3. As reported earlier, the change in

---

Table 2  Results

<table>
<thead>
<tr>
<th>Variables</th>
<th>Baseline model</th>
<th>+PI</th>
<th>+Interactions</th>
<th>Baseline model</th>
<th>+PI</th>
<th>+Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFP</td>
<td>-1.111 (0.148)**</td>
<td>0.195 (0.085)**</td>
<td></td>
<td>0.211 (0.082)**</td>
<td></td>
<td>0.383 (0.091)**</td>
</tr>
<tr>
<td>ScheduleCompression</td>
<td>-0.797 (0.074)**</td>
<td>-0.221 (0.043)**</td>
<td></td>
<td>-0.232 (0.042)**</td>
<td></td>
<td>-0.242 (0.044)**</td>
</tr>
<tr>
<td>TeamSize</td>
<td>-0.234 (0.102)**</td>
<td>-0.128 (0.057)**</td>
<td></td>
<td>-0.112 (0.055)**</td>
<td></td>
<td>-0.114 (0.055)**</td>
</tr>
<tr>
<td>TeamDispersion</td>
<td>-0.349 (0.074)**</td>
<td>0.062 (0.024)**</td>
<td></td>
<td>0.073 (0.024)**</td>
<td></td>
<td>0.121 (0.05)**</td>
</tr>
<tr>
<td>PMTaskFamiliarity</td>
<td>0.274 (0.106)**</td>
<td>0.078 (0.071)**</td>
<td></td>
<td>0.076 (0.068)**</td>
<td></td>
<td>0.106 (0.069)**</td>
</tr>
<tr>
<td>TeamTaskFamiliarity</td>
<td>0.45 (0.086)**</td>
<td>0.067 (0.025)**</td>
<td></td>
<td>0.072 (0.025)**</td>
<td></td>
<td>0.122 (0.054)**</td>
</tr>
<tr>
<td>TeamFamiliarity</td>
<td>0.231 (0.074)**</td>
<td>0.105 (0.043)**</td>
<td></td>
<td>0.106 (0.042)**</td>
<td></td>
<td>0.162 (0.047)**</td>
</tr>
<tr>
<td>ClientExp</td>
<td>-0.128 (0.097)</td>
<td>0.152 (0.067)**</td>
<td></td>
<td>0.149 (0.064)**</td>
<td></td>
<td>0.214 (0.065)**</td>
</tr>
<tr>
<td>Contract = TM</td>
<td>-0.383 (0.203)*</td>
<td>-0.208 (0.122)*</td>
<td></td>
<td>-0.205 (0.117)*</td>
<td></td>
<td>-0.169 (0.095)*</td>
</tr>
<tr>
<td>ProjectPLP = Low</td>
<td>-0.738 (0.146)**</td>
<td>-0.214 (0.084)**</td>
<td></td>
<td>-0.21 (0.081)**</td>
<td></td>
<td>-0.224 (0.078)**</td>
</tr>
<tr>
<td>ProjectType = M</td>
<td>0.444 (0.148)**</td>
<td>0.204 (0.086)**</td>
<td></td>
<td>0.248 (0.083)**</td>
<td></td>
<td>0.21 (0.071)**</td>
</tr>
<tr>
<td>PMWorkEx</td>
<td>0.038 (0.087)</td>
<td>0.205 (0.061)**</td>
<td></td>
<td>0.206 (0.056)**</td>
<td></td>
<td>0.26 (0.054)**</td>
</tr>
<tr>
<td>PMRating</td>
<td>-0.126 (0.09)</td>
<td>0 (0.066)</td>
<td></td>
<td>-0.002 (0.061)</td>
<td></td>
<td>-0.003 (0.058)</td>
</tr>
<tr>
<td>PriorProjectOrg</td>
<td>0.206 (0.067)**</td>
<td>-0.109 (0.04)**</td>
<td></td>
<td>-0.112 (0.038)</td>
<td></td>
<td>-0.077 (0.037)</td>
</tr>
<tr>
<td>PI</td>
<td>0.44 (0.09)**</td>
<td>0.468 (0.066)**</td>
<td></td>
<td>0.469 (0.064)**</td>
<td></td>
<td>0.505 (0.064)**</td>
</tr>
<tr>
<td>LogFPXPI</td>
<td>0.268 (0.093)**</td>
<td>0.447 (0.094)**</td>
<td></td>
<td>0.447 (0.094)**</td>
<td></td>
<td>0.447 (0.094)**</td>
</tr>
<tr>
<td>ScheduleCompressionXPI</td>
<td>0.083 (0.008)**</td>
<td>0.256 (0.041)**</td>
<td></td>
<td>0.256 (0.041)**</td>
<td></td>
<td>0.256 (0.041)**</td>
</tr>
<tr>
<td>TeamSizeXPI</td>
<td>0.22 (0.097)**</td>
<td>0.139 (0.044)**</td>
<td></td>
<td>0.139 (0.044)**</td>
<td></td>
<td>0.139 (0.044)**</td>
</tr>
<tr>
<td>TeamDispersionXPI</td>
<td>0.096 (0.074)</td>
<td>-0.18 (0.051)**</td>
<td></td>
<td>-0.18 (0.051)**</td>
<td></td>
<td>-0.18 (0.051)**</td>
</tr>
<tr>
<td>PMTaskFamiliarityXPI</td>
<td>-0.149 (0.103)**</td>
<td>0.135 (0.088)**</td>
<td></td>
<td>0.135 (0.088)**</td>
<td></td>
<td>0.135 (0.088)**</td>
</tr>
<tr>
<td>TeamTaskFamiliarityXPI</td>
<td>-0.268 (0.093)**</td>
<td>0.153 (0.056)**</td>
<td></td>
<td>0.153 (0.056)**</td>
<td></td>
<td>0.153 (0.056)**</td>
</tr>
<tr>
<td>TeamFamiliarityXPI</td>
<td>-0.199 (0.084)**</td>
<td>0.522 (0.06)**</td>
<td></td>
<td>0.522 (0.06)**</td>
<td></td>
<td>0.522 (0.06)**</td>
</tr>
<tr>
<td>ClientExpXPI</td>
<td>-0.055 (0.134)</td>
<td>-0.263 (0.084)**</td>
<td></td>
<td>-0.263 (0.084)**</td>
<td></td>
<td>-0.263 (0.084)**</td>
</tr>
<tr>
<td>dVendorClient</td>
<td>0.732 (0.241)**</td>
<td>5.173 (0.23)**</td>
<td>0.477 (0.202)**</td>
<td>5.173 (0.23)**</td>
<td>0.477 (0.202)**</td>
<td>5.173 (0.23)**</td>
</tr>
</tbody>
</table>

Model fit statistics

<table>
<thead>
<tr>
<th></th>
<th>Cost performance</th>
<th>Client satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N)</td>
<td>530</td>
<td>530</td>
</tr>
<tr>
<td>(R^2)</td>
<td>0.456</td>
<td>0.503</td>
</tr>
<tr>
<td>Change in (R^2)</td>
<td>0.456</td>
<td>0.048</td>
</tr>
<tr>
<td>(F) test for change in (R^2)</td>
<td>41.743</td>
<td>22.249</td>
</tr>
<tr>
<td>(p) value of (F) test</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(p) value of (F) test</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Note. Standard errors in parentheses.

\(*\)Significant at 10%; \(**\)significant at 5%; \(**\)significant at 1%.

\(|p| < 0.05\).
Discussion of Results

In the following, we discuss the results for the main effects of PMs’ PI, as well as the moderating effects of the project context on the relationship between PMs’ PI and project outcomes.

First, the significance of the main effects of PI in our models of project performance emphasizes how important PMs’ PI is for cost performance and client satisfaction in the software offshore outsourcing setting. Our regression estimates show that for a typical project, if PMs’ PI was half a standard deviation above the average, the project could have saved $75,900 in project costs and had an increase of 0.28 (on a scale of 1 to 7) for client satisfaction; note that a typical project costs about $1.2 million and has a client satisfaction rating of 5.68. Clearly, the impact of PMs’ PI on project performance is meaningful as well as significant, given that we are controlling for many other factors relating to project performance and that we are also studying a CMM Level 5 organization that has implemented best practices and frameworks for project management. The benefits of PMs’ PI for client satisfaction are particularly important, given the vendor’s paramount need to attract new and continued client engagements.

Second, our results offer novel and nuanced insights on how elements of the project context affect the value of PMs’ PI for project performance. To help visualize the interaction effects, we present our results graphically. Specifically, we find that PMs’ PI is more helpful for more complex projects, where the project manifests increased technological or organizational complexity (Baccarini 1986). Figures 2(a)–2(c) show that the positive effect of PMs’ PI on project performance is stronger for projects that are higher in complexity for both cost performance and client satisfaction; thus having PMs with higher levels of PI is beneficial for projects that have larger software size, greater schedule compression, or larger teams. PMs’ PI tempers the negative effects of complexity on project performance. We show that at higher levels of complexity, these benefits are even greater. For example, as Figure 2(b) implies, for a project with high schedule compression (where costs are in the range of $3 to 5 million), an increase in PMs’ PI from low to high levels would yield a cost savings of approximately $1,070,000, and client satisfaction would increase by 0.93. Figure 2(d), however, suggests that team dispersion is substitutive with PMs’ PI in affecting client satisfaction, contrary to our ex ante prediction that projects with greater team dispersion would benefit more from PMs with greater PI; our results imply that expending more effort onsite—by deploying more team members at the client site—may in fact help the team develop more familiarity and understanding with the client, reducing the likelihood of critical incidents. Such projects rely less on PMs’ PI and consequently, the marginal benefit is less from PMs’ PI. The beneficial effects of PMs’ PI are especially salient for client satisfaction outcomes: Figures 2(a)–2(c) suggest that even for highly complex projects, client satisfaction can equal or exceed levels achieved for low complexity projects, if they are managed by a PM with high PI.

Our results also show that projects with low task or stakeholder familiarity (Goodman and Leyden 1991) benefit even more from PMs’ PI. As shown in Figures 3(a) and 3(d), higher levels of PMs’ PI are more helpful for projects with lower team task familiarity and lower team familiarity. There is partial support for our hypotheses on the interactions between PMs’ PI and PMs’ task and client familiarity. As shown in Figure 3(b), PMs’ PI can substitute for PMs’ task familiarity with respect to cost performance in that higher levels of PMs’ PI are more helpful for projects with lower PMs’ task familiarity, but the effect is not significant for client satisfaction. Figure 3(c) depicts that for client satisfaction, PMs’ PI can substitute for PMs’ client familiarity, such that higher levels of PMs’ PI are more helpful for projects with lower PMs’ client familiarity. Our results indicate that even for projects with low levels of task or stakeholder familiarity, having a PM with greater PI in charge is associated with a level of client satisfaction that equals or exceeds that for projects with higher familiarity.

In sum, our results indicate that PMs’ PI is beneficial for cost performance as well as client satisfaction, and that the benefits of PI are even higher for more complex or less familiar projects.

5. Conclusion

Successfully managing software projects to deliver the right solutions on time and within budget is difficult. Software offshore outsourcing poses special
management challenges given the diversity of tasks and stakeholders dispersed geographically and across organizations (Carmel 1999, Levina and Vaast 2008). The information asymmetries inherent in such projects can lead to gaps between the project’s information processing needs and capabilities and thus increase the likelihood of unforeseen critical incidents that must be resolved effectively for successful project outcomes.

Our study introduces PMs’ PI as the contextual ability to resolve project related problems that involve extreme situations, which often do not have readily available solutions (Joseph et al. 2010, Wagner and Sternberg 1985). We argued that, as the primary decision makers in these projects, PMs need PI to resolve critical incidents related not only to task, self, and career, but also to the numerous project stakeholders. Our study
identifies the project context that is likely to increase a project’s information constraints—to wit, projects that have high complexity or low familiarity—and hence benefit most from PMs’ PI. We now discuss the contributions and practical implications of this study as well as limitations and future research.

Contributions
Our study contributes to the extant literature in several ways. First, we contribute to the project management literature by demonstrating the project performance effects of PI, a capability that is situational, contextual, and embedded in the PM. This literature has emphasized project management skills (Kirsch 2000, Napier et al. 2009), experience in technology, domain, and methodology (Duncan 1996, Thite 1999), the use of best practices (Gopal et al. 2002), and risk management frameworks (Barki et al. 2001, Wallace et al. 2004). This literature also tends to take a “best practices” perspective, even as recent research on project “tailoring” suggests the potential performance benefits of adjusting standard practices to different contexts (e.g., Slaughter et al. 2006). Although standard project management practices and skills are useful for managing routine challenges, they are ill-equipped to address the dynamic information needs of diverse project contexts typical in software offshore outsourcing that lead to unexpected, nonroutine situations. Our findings offer PMs’ PI as an additional “capability” that can be flexibly deployed where it is needed most, aimed at fitting solutions to problems that occur in a particular project context.

Second, our study makes a significant contribution to the literature on offshore outsourcing, which has examined the effects of contract type and contractual governance mechanisms, transaction cost economies, and principal agent models on project outcomes (e.g., Gopal et al. 2003), but does not substantiate the central role played by PMs in ensuring the success of such initiatives. Recent research has explored cultural and social leadership aspects of PMs in the context of software outsourcing, but has found relatively little empirical support on how these aspects impact project outcomes (Rai et al. 2009, Chua et al. 2012). Using archival data
and rigorous empirical analysis, we identify a factor (PMs’ PI) that, after controlling for contractual, process, and other project attributes, contributes significantly to project performance in this setting.

Finally, we extend the PI literature (Wagner and Sternberg 1985, and more recently Joseph et al. 2010) by conceptualizing and quantifying PMs’ PI and identifying its efficacy in the context of software offshoring projects. The literature on PI has focused on individual level outcomes but has largely ignored the effects of PI on outcomes at a more aggregate level. By recognizing the need for PMs’ PI in the context of software outsourcing, we provide empirical support to show that this capability is effective for project level outcomes. We also demonstrate how complexity and familiarity, the two opposite attributes that affect information constraints in a project, shape the contribution of PMs’ PI to project outcomes.

Practical Implications
Although PMs with high levels of PI are desirable for all projects, such PMs are not always available. Thus, our findings have important implications for senior managers, as they select PMs for projects and advise and groom potential candidates for PM positions. Given that PMs with PI are a scarce resource, it is judicious to allocate these PMs to the projects that will benefit from them the most. In terms of selection, our results suggest that there may be significant benefits in terms of higher project performance from assigning a PM to a project based on the levels of the PMs’ PI, and the project’s context, that is, the level of project complexity and familiarity. For example, as we have discussed, projects that are anticipated to have high complexity or low familiarity would benefit more from having a PM with higher PI in charge than would projects of lower complexity or higher familiarity. Thus, for a high complexity or low familiarity project, senior managers may find it worthwhile to assign it to a PM with high levels of PI or wait for one to become available if the PM is not immediately available to lead the project. Conversely, for a project with less complexity or more familiarity, while high PMs’ PI is still desirable, the marginal benefits may be lower. Thus, senior managers may assign a PM with lower levels of PI to the project without severely affecting performance, because these PMs can rely on available processes and frameworks for effective project management. Thus, a more nuanced PM selection strategy that tailors PMs’ PI to the project context enables the vendor firm to achieve better project outcomes.

In terms of training candidates for PM positions, our findings suggest that project management experience, per se, may not be indicative of an individual’s true potential as a PM, in that project management experience may not always be commensurate with greater PI. This finding is consistent with Wagner and Sternberg’s (1985) contention that work experience and PI are not the same. In our study, PMs’ PI is correlated only modestly with PMs’ project management experience (0.13), and the effects of PMs’ project management experience on project outcomes are significantly lower than those of PMs’ PI.13 By enabling contextual management of the project, leads to better project outcomes. Despite the process oriented culture prevalent at many software outsourcing vendors, there are differences in project outcomes and hence it may be necessary for senior management to provide tailored training to PMs to help them acquire the PI they need to effectively manage more challenging projects. Although an intangible and contextual ability such as PI may not be taught effectively in a traditional instructional mode, Wagner and Sternberg (1990) suggest that successful managers can share with others, either through direct instruction or mentorship, (a) the rules of thumb they use in day-to-day situations and (b) the strategies for acquiring PI. Organizations can also incorporate critical incident management simulation and role play to enhance PI. It could also be possible to evaluate how successful PMs with high PI resolve critical situations, and use those solutions as best practices. PMs with high PI can also be considered experts who are sought out when critical incidents occur in a project, with adequate compensation, of course. As software vendors move toward true competency development, this kind of specific training may provide benefits in terms of better cost structure and repeat business but also in increasing their specific human capital assets, leading to better retention of existing capabilities. Our findings are thus particularly relevant to domains like software offshoring outsourcing that have severe information constraints and suffer from a high likelihood of critical incidents.

Limitations and Future Research
As does all research, our study has limitations and poses opportunities for further research. The data was collected from a single vendor, though the vendor and the projects we chose are representative of the software outsourcing projects that are currently being executed. That said, our data collection strategy has several critical strengths in that the focus on one organization controls for organizational factors that would otherwise confound our results. In addition, we were able to collect multiple projects for each project manager; only with such data can we distinguish between the PMs’ PI and project characteristics in identifying their respective effects on project outcomes. Although a single site

---

13 When comparing coefficients, we find that PMs’ PI has a far greater impact on both cost performance and client satisfaction than their project management experience (CostPerformance: χ²(1) = 12.02, p < 0.001; ClientSatisfaction: χ²(1) = 29.59, p < 0.001).
gives us better control in terms of model identification, our results may be indicative of practices prevalent at the vendor site and organizational peculiarities. Our findings are most likely to generalize to large software outsourcing vendors, and may not apply as readily to in-house software projects. Further, as it was practically infeasible to collect measures of PMs’ PI before the start of each project, we have a post-hoc measure of PI. Although we control for relevant project management experience for each PM in our analysis, the post hoc measure of PMs’ PI may reflect a progressive learning bias. However, our measure of PMs’ PI is likely to provide a conservative view of the value of PI, and thus understates the true effects of PMs’ PI.14

Our study offers several interesting directions for future research. One direction concerns the measurement of PI. We adopted the approach used in the literature (e.g., Wagner and Sternberg 1985, Joseph et al. 2010) to measure and evaluate PI. Although this approach is well grounded, the assessment of critical incidents is labor intensive. Future work could focus on measuring PI by automating the assessment of critical incidents. Another possibility for future research is to examine other determinants of project performance and their relationship with PMs’ PI. Given the diverse cultural backgrounds of project stakeholders characteristic of software outsourcing, it may also be useful to understand how cultural intelligence (Earley and Ang 2003) relates to PI. We focus on complexity and familiarity as the two pertinent aspects that affect information constraints in a project; future research could examine nuanced factors such as requirements volatility (Barry et al. 2006, Nidumolu 1995) and how PI can help projects with higher requirements volatility. Finally, our study examines the role of PMs’ PI in software offshore outsourcing. Future studies in other settings and for other projects would be useful to generalize and validate our results.

Supplemental Material
Supplemental material to this paper is available at http://dx.doi.org/10.1287/isre.2014.0523.

Acknowledgments
The authors thank the seminar and conference participants at the Indian School of Business, Carnegie Mellon University, the University of Washington, New York University, Purdue University, the 2008 International Conference on Information Systems, and the 2007 INFORMS Annual Meeting (among others) for their helpful comments and suggestions on earlier versions of this paper. The authors thank the senior editor and the review team for their careful reading of the paper and the insightful comments. The authors thank CyLab at Carnegie Mellon University for the financial support needed for this research. The authors also thank everyone at their research site for generously contributing time, data, and effort toward the execution of this research. All errors remain those of the authors.

References

14 As a robustness check, we performed a cross-sectional analysis in which we estimated separate models for the first, second, and subsequent projects managed by each PM. We find that the effect of PI is consistent over time and precludes any learning effect. The results of this analysis are available from the authors upon request.


