12th Pre-ICIS International Research Workshop on Information Technology Project Management (IRWITPM 2017) 
Workshop Proceedings

12/9/2017
AIS Special Interest Group on Information Technology Project Management
Table of Contents

Workshop Welcome ..................................................................................................................................................3
Workshop Committee ...............................................................................................................................................3
SIG ITProjMgmt Officers .........................................................................................................................................3
Sponsor Thanks ......................................................................................................................................................4

Completed Research

* The Role of Time Pressure in Software Projects: A Literature Review and Research Agenda by Dirk Basten ...........................................................................................................................................5

Team Performance in Agile Software Development Projects: The Effects of Requirement Changes, Time Pressure, Team Diversity and Conflict by Phil Diegmann and Christoph Rosenkranz .........................................................................................18


Research in Progress

Implementing Strategy Through IS Projects: A Theory Building Literature Review by Fred Niederman and Kathy Chudoba .........................................................................................................................................46

Temporality in Information Systems Development (ISD) Research: A Systematic Literature Review by Mairead O Connor, Denis Dennehy, Kieran Conboy .........................................................................................................................55

Note: * marks best paper award.
WORKSHOP WELCOME

I would like to personally welcome each of you to Seoul, South Korea for the 12th International Research Workshop on IT Project Management (IRWITPM 2017) sponsored by the AIS Special Interest Group for Information Technology Project Management (SIGITProjMgmt). This is our twelfth consecutive annual workshop and I am so excited that we can continue tradition of excellence as we promote, encourage, and discuss research in the domain of IT project management.

I hope that you enjoy the workshop proceedings that cover a wide range of topics relating to IT project management. Technology continues to evolve and provides both opportunities and challenges in relation to our work practices, thus providing many opportunities for research. Although technology changes, one thing stays constant and that is people and processes. People are a critical component to ensuring processes are in place to support work practices.

I would like to thank each of you for attending our workshop and bringing your expertise. I also want to send my gratitude to the workshop authors, reviewers, organizers, and sponsors (Project Management Institute) - your effort makes this event possible. My sincere thanks to all for your efforts and engaging with this AIS Special Interest Group.

Dawn Owens, University of Texas at Dallas, SIGITProjMgmt President

WORKSHOP COMMITTEE

Dawn Owens, University of Texas at Dallas (Workshop Program Co-Chair)
Stacie Petter, Baylor University (Workshop Program Co-Chair)
Alanah Mitchell, Drake University (Workshop Program Co-Chair)

SIG ITPROJMGMT OFFICERS

Dawn Owens, University of Texas at Dallas (President)
Michael Cuellar, Georgia Southern University (Secretary)
Radu Vlas, University of Houston – Clear Lake (Treasurer)
Mohammad Moenini Aghkariz, University of Sussex (Communications and Publicity Chair)
Cecil Chua, The University of Auckland (Membership and Community Relations Chair)
Deepak Khazanchi, University of Nebraska at Omaha (Founder)
The Role of Time Pressure in Software Projects: A Literature Review and Research Agenda

Dirk Basten
University of Cologne
basten@wiso.uni-koeln.de

ABSTRACT

The finding that deadlines affect work in organizational settings holds particularly true for software projects, which are usually conducted under time pressure. While the role of time pressure in software projects has been extensively studied, the findings yielded are diverse. Some authors report a positive relationship between time pressure and software project outcomes, while others find it to be negative or to follow an inverted U-shape pattern. Since many aspects concerning time pressure remain unexplained and its relationship with project outcomes is more complex than it might seem at first glance, we synthesize pertinent research to develop a research agenda aimed at improving the understanding in this domain. Our literature review shows a variety of time pressure conceptualizations, research approaches, and research contexts. The results reveal an inconsistent picture of time pressure’s impact on software projects. Our research agenda includes five themes we deem beneficial to consider in future research.

Keywords

Software projects, software development, time pressure, project management, literature review, research agenda.

INTRODUCTION

It is commonly acknowledged that deadlines affect work in organizational settings (Marks, Mathieu and Zaccaro, 2001). This is particularly the case for software development, where the ‘iron triangle’ is traditionally used to assess project success (Atkinson, 1999, Pinto, 2004). The three criteria of the ‘iron triangle’—time, budget and quality—are closely interrelated. Empirical evidence indicates that, in most software projects, effort estimates in terms of time or budget are typically unrealistically low (Basten and Sunyaev, 2014), making schedule and budget overruns a common issue (Grimstad, Jørgensen and Moløkken-Østvold, 2006, Kappelman, McKeeman and Zhang, 2006, Mukhopadhyay, Vicinanza and Prietula, 1992). Such findings apply to both sequential (Maruping, Venkatesh, Thatcher and Patel, 2015) and agile (Malgone, Collins and Hevner, 2014) projects. Time pressure that results from overly optimistic estimates is usually associated with negative outcomes, such as reduced software quality (Mäntylä and Itkonen, 2013).

The role of time pressure in software projects has been the subject of extensive research (e.g., Austin, 2001, Banker, Datar and Kemerer, 1987, Mäntylä, Petersen, Lehtinen and Lassenius, 2014, Maruping et al., 2015). However, findings yielded by these studies are inconclusive, and this lack of consistency can be attributed to the absence of a generally accepted definition of the concept (Hwang, 1994). Evidence of both negative and positive impact of time pressure on work performance has been reported in extant literature (Maruping et al., 2015). Since under time pressure software developers simply work faster rather than better (DeMarco, 1982), having unrealistic deadlines can lead to software quality reductions (Mäntylä et al., 2014). Conversely, quality of decisions has been found to increase when time-dependent incentives are given (Kocher and Sutter, 2006). Additionally, several authors posit that the nature of this link is dependent on the degree of time pressure (Maruping et al., 2015, Nan and Harter, 2009). Yet, despite extensive research in this field, many aspects of time pressure’s impact on software projects remain unexplained (Siau, Long and Ling, 2010). Indeed, the relationship between time pressure and software projects is more complex than it might seem at first glance (Mäntylä and Itkonen, 2013). Thus, in order to improve the understanding, we aim to elucidate the role of time pressure in software projects by answering the following research questions:

What is the state of research concerning the role of time pressure in software projects?

How should the role of time pressure in software projects be addressed in future research?

In order to answer these questions, we conducted a review of literature held in six major scientific databases, focusing on publications that address the role of time pressure in software projects. In analyzing the work reported within, we
identified a variety of time pressure conceptualizations, research approaches, and research contexts. Considering existing insights into the role of time pressure in software projects, we found an inconsistent picture concerning the effects of time pressure and respective practical implications. Thus, in order to reduce these discrepancies and assist authors of future works in this field, we developed a research agenda comprising five major themes that should be considered to advance the understanding of time pressure’s role in software projects.

This paper is structured as follows. In the next section, we explain the core concepts pertaining to this investigation and report insights gained through the review of extant research. We then describe the strategy we adopted for the identification and analysis of publications concerning time pressure’s role in software projects. Subsequently, we present the results of the literature review pertaining to the conceptualization of time pressure, research approaches and contexts used, and the observed effects of time pressure. We then discuss the findings and develop an agenda for future research. Finally, we conclude the paper by considering the study limitations and practical implications of our findings.

THEORETICAL BACKGROUND

Software Projects

A project is defined as “a temporary endeavor undertaken to create a unique product, service, or result” (Project Management Institute, 2013, p. 3). While software has been defined as “computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system” (Linberg, 1999, p. 179), software development is the process of creating software that realizes and fulfills customer expectations (Bourque and Fairley, 2014). Combining these definitions, software projects are projects in above terms for developing (or extending/adapting) software. While a plethora of development approaches exists, researchers commonly differentiate between sequential and agile development as the two extremes (e.g., Dybå and Dingsøyr, 2008, Palmquist, Lapham, Miller, Chick and Ozkaya, 2013). Table 1 juxtaposes the key characteristics of both approaches, which are described in more detail below.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Sequential Development</th>
<th>Agile Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>sequential</td>
<td>iterative</td>
</tr>
<tr>
<td>Activities (e.g., design, implementation)</td>
<td>one-time</td>
<td>in each iteration</td>
</tr>
<tr>
<td>Customer Involvement</td>
<td>mainly during requirements engineering and testing</td>
<td>in each iteration</td>
</tr>
<tr>
<td>Documentation</td>
<td>extensive</td>
<td>limited, tailored to the project</td>
</tr>
</tbody>
</table>

Table 1. Juxtaposition of Sequential and Agile Development Approaches (based on Palmquist et al., 2013)

Sequential development is the traditional approach to developing software (Dybå and Dingsøyr, 2008). Traditional approaches stem from the engineering discipline, where solutions are claimed to exist for every problem and problems to be solved are presumed to be fully specifiable. Traditionalists thus put emphasis on extensive planning and codified processes, as this is deemed to make software development efficient and predictable (Boehm, 2002). In the traditional perspective, software development is a sequential (i.e., non-iterative) process that commences with requirements engineering, followed by system design and implementation, ending with testing and integration, and maintenance (Palmquist et al., 2013). To proceed from one phase to the next, formal review and approval is required. Thus, the development process is accompanied by extensive documentation and customers are typically involved in determining requirements and testing the software.

Agile development is “both a philosophy and an umbrella term for a collection of methods or approaches that share certain common characteristics” (Palmquist et al., 2013, p. 9) rather than a single method. Common characteristics include (Palmquist et al., 2013):

- Requirements are assumed to change.
- To counteract requirements uncertainty, software is created in short iterations, which all include the activities of analysis, design, coding, and testing.
- Documentation is typically tailored to a project.
- Customers are involved in each iteration to approve software and decide how to proceed.
Rather than emphasizing processes, agile development approaches focus on people, their creativity and their capacity to cope with the challenge of uncertainty (Dybå and Dingsøyr, 2008). Feedback and change are the key aspects of agile development approaches, which “embrace, rather than reject, higher rates of change” (Williams and Cockburn, 2003, p. 39). The practices on which agile software development approaches are based have been created by experienced practitioners (Ågerfalk and Fitzgerald, 2006). Common approaches to agile software development include Extreme Programming (Beck, 2000) and Scrum (Schwaber and Beedle, 2002).

**Time Pressure in Software Projects**

Since software development is typically project-based, the requisite endeavors take place under clearly defined time and budget constraints (Bourque and Fairley, 2014). Managers of software projects thus need to consider the three indices—time, budget, and quality—which are commonly referred to as ‘iron triangle’ (Atkinson, 1999) or ‘triple constraint’ (Pinto, 2004). The three indices are closely related since they typically affect each other. For instance, restricting the budget is likely to reduce quality and time available to develop a software. In practice, when planning projects, estimates of effort in terms of time and budget are typically unrealistically low (Basten and Sunyaev, 2014). Consequently, overruns of project schedule and budget occur (Grimstad et al., 2006, Kappelman et al., 2006, Mukhopadhyay et al., 1992), resulting in time pressure.

Time pressure, which is the perception that time available to complete a task is scarce in relation to the demands of the task (Cooper, Dewe and O'Driscoll, 2001, Kelly and McGrath, 1985), is common in organizational settings (Gersick, 1988, Gevers, Rutte and van Eerde, 2006, Waller, Zellmer-Bruhn and Giambatista, 2002). Since “factors such as project deadlines […] dictate many aspects of team functioning, including the strategies that are employed [and] the pace of activities […] in order for the teams to perform successfully” (Marks et al., 2001, p. 359), it is widely acknowledged that time pressure affects employee behavior and performance.

While authors of several studies have addressed the role of time pressure in software projects (e.g., Austin, 2001, Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Maruping et al., 2015, Shah, Harrold and Sinha, 2014), their findings concerning the relationship between time pressure and project performance are inconclusive (Maruping et al., 2015). For example, time pressure has been found to exert both positive and negative impact on work performance, while an inverted U-shape relationship has been reported at the individual level in some cases. Recent empirical work has revealed that time pressure in software development can improve efficiency (as measured in, for example, more defects found per time-unit and greater scores on test cases) in test case development (Mäntylä et al., 2014).

An extensive body of studies exploring the role of time pressure in software projects presently exists. However, the explanatory power and theoretical contribution of this stream of literature remains inconclusive. Additionally, extant studies differ regarding the depth to which the role of time pressure in software projects is assessed. While some authors treat time pressure as one of many variables with the potential to explain various phenomena, such as developers’ attitude that results from time pressure in projects (Sojer, Alexy, Kleinknecht and Henkel, 2014), others have a clear focus on time pressure as the study concept (e.g., Austin, 2001). Moreover, a systematic synthesis of such studies is presently lacking, indicating that an overview and analysis of these studies is needed to improve the understanding of time pressure’s role in software projects. Insights into the research approaches applied and contexts studied would also be highly beneficial in providing guidance for future research in this field. Additionally, we posit that such a synthesis would provide a much-needed structure to the previous studies and would elucidate important, yet under-researched, phenomena in this regard.

**RESEARCH APPROACH**

In order to assess the role of time pressure in software projects and thus answer the two research questions guiding this investigation, we conducted a concept-driven literature review (Webster and Watson, 2002) and applied a two-phase approach to identify and analyze relevant literature sources.

**Identification and Selection of Literature**

We chose to follow a tollgate approach for the selection of publications (see Figure 1; cf. Afzal, Torkar and Feldt, 2009). In Step 1, we identified potentially relevant publications by conducting systematic searches in the following databases: ACM Digital Library, AIS Electronic Library (AISel), EBSCOhost, IEEE Xplore, ProQuest and ScienceDirect. For our search, we relied on a query within the search fields title, abstract, and keywords for each
As time pressure is not a new phenomenon in software development, we did not restrict the search period. Nonetheless, to assess our search process, we compared the identified publications with a small set of known studies (i.e., Austin, 2001, Mäntylä and Itkonen, 2013, Maruping et al., 2015). The findings yielded justify our choice of search process (cf. Kitchenham, Mendes and Travassos, 2007) because all articles known beforehand were among the identified publications. Our search approach yielded 36,729 results. We reduced the set of publications to the most relevant ones by only including articles published in journals in the Journal Ranking of the Association for Information Systems (AIS). Additionally, we considered selected conferences for the analysis. The selection of outlets resulted in a set of 1,923 articles potentially relevant to our research purpose. In order to refine our sample, we read titles and abstracts to determine the work’s relevance to our research objective. Finally, and based on full reading of the text, we excluded editorials (e.g., Glass, 2004), research-in-progress papers (Malgone et al., 2014), and studies in which time pressure is treated as a problem in software projects without analyzing its impact (Cao and Ramesh, 2008). This strategy resulted in a set of 11 publications matching our criteria.

Figure 1. Selection of Publications

---

1 Recently, the AIS removed the journal ranking from its website. It is available at:

2 We included the following conference proceedings in our review: Americas Conference on Information Systems (AMCIS), European Conference on Information Systems (ECIS), Hawaii International Conference on System Sciences (HICSS), International Conference on Information Systems (ICIS), and International Conference on Software Engineering (ICSE).

---

Basten

Time Pressure in Software Projects

database. We used a combination of the terms “time pressure” AND “software”, including relevant synonyms for both terms. We adopted the following search pattern and adapted it to match the syntax of each database:

("deadline pressure" OR "time pressure" OR "time restriction" OR "time limitation" OR "schedule pressure" OR "time constrain") AND (program OR software OR project OR "information technology" OR "information system")

We included the following conference proceedings in our review: Americas Conference on Information Systems (AMCIS), European Conference on Information Systems (ECIS), Hawaii International Conference on System Sciences (HICSS), International Conference on Information Systems (ICIS), and International Conference on Software Engineering (ICSE).
In Step 2, we performed backward and forward searches (Webster and Watson, 2002) to complement the set of publications. Our backward search involved analysis of the references listed in the 11 publications that emerged from the initial search process. For searching forward, we used Google Scholar (http://scholar.google.com). While the backward search did not yield further relevant publications, we added two publications to our set based on the forward search.

**Data Analysis**

We analyzed the final set of 13 publications by using a structured coding scheme (Flick, 2009, Miles and Huberman, 1994). The coding scheme consists of the following three code families (see Table 2 for respective codes and definitions): (1) Codes related to the conceptualization of time pressure, addressing the definition of time pressure and related information that the authors provide in the corresponding articles; (2) The code family concerning the research approach and context, addressing studies’ research focus, research methods, and whether the studies are based on empirical data; and (3) Codes related to time pressure’s effects, which include information about target variables, results and respective implications.

<table>
<thead>
<tr>
<th>Code Family</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptualization of time pressure</td>
<td>Concept label</td>
<td>Is the concept referred to as time pressure, deadline pressure, schedule pressure, etc.?</td>
</tr>
<tr>
<td></td>
<td>Definition</td>
<td>Is the concept explicitly defined? How do the authors define / conceptualize or operationalize / measure the concept?</td>
</tr>
<tr>
<td></td>
<td>Measurement</td>
<td>Is time pressure seen as objective or as a subjective perception?</td>
</tr>
<tr>
<td>Level</td>
<td>Research focus</td>
<td>What is the study’s focus?</td>
</tr>
<tr>
<td>Perspective</td>
<td>Perspective(s) from which time pressure is considered (e.g., managers, developers, testers).</td>
<td></td>
</tr>
<tr>
<td>Research method</td>
<td>What research method do the authors employ (e.g., conceptual, literature review, case study research, survey)?</td>
<td></td>
</tr>
<tr>
<td>Empirical data</td>
<td>Is the study based on empirical data? Is secondary data included?</td>
<td></td>
</tr>
<tr>
<td>Effects of time pressure</td>
<td>Target variable</td>
<td>What target variables are considered (e.g., quality, motivation, effectiveness, efficiency)?</td>
</tr>
<tr>
<td>Results</td>
<td>What type of effect has been shown for time pressure on the target variable(s)?</td>
<td></td>
</tr>
<tr>
<td>Implications</td>
<td>What implications (if any) are provided based on the results?</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Structured Coding Scheme

For the development of the research agenda, we relied on thematic analysis, employed in qualitative studies when the aim is to identify, analyze and report patterns within data (Braun and Clarke, 2006). By following an inductive approach, we attempted to capture important aspects identified in our literature review in relation to our second research question. Thereby, a theme “captures something important about the data in relation to the research question, and represents some level of patterned response or meaning within the data set” (Braun and Clarke, 2006, p. 82).

**RESEARCH STATE**

Table 3 provides an overview of the analyzed articles in alphabetical order, while our results are structured in three subsections, pertaining to conceptualization of time pressure, research approach and context, and effects of time pressure.

**Conceptualization of Time Pressure**

We present the results related to the labelling employed in analyzed studies for the concept time pressure, along with its definition and the level from which time pressure is addressed.

While authors of several studies in our sample refer to the concept of scarcity of time available to complete a task, ‘time pressure’ as the sole label for this concept is used in seven studies only (Austin, 2001, Deak, Stålhane and Sindre, 2016, Lohan, Acton and Conboy, 2014, Mäntylä et al., 2014, Maruping et al., 2015, Pennington and Tuttle, 2007, Sojer et al., 2014). Others refer to deadline pressure (Banker et al., 1987), time constraints (Do, Mirarab, Tahvildari...
and Rothermel, 2010), or schedule pressure (Nan and Harter, 2009). Finally, some authors use time pressure synonymously with terms such as time restriction (Mäntylä and Itkonen, 2013), deadline pressure (Shah et al., 2014) or time availability constraints/limitations (Topi, Valacich and Hoffer, 2005).

<table>
<thead>
<tr>
<th>Study</th>
<th>Time Pressure Focus of the Study</th>
<th>Research Method</th>
<th>Empirical Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin (2001)</td>
<td>Development of a game-theory model to explain the effect of time pressure on software developers’ behavior towards (not) taking shortcuts</td>
<td>Conceptual</td>
<td>No</td>
</tr>
<tr>
<td>Banker et al. (1987)</td>
<td>Analysis of the effect of time pressure on the productivity of software maintenance</td>
<td>Survey</td>
<td>Yes</td>
</tr>
<tr>
<td>Deak et al. (2016)</td>
<td>Analysis of how professional software testers can be motivated and exploration of the respective policies to be implemented in software development projects</td>
<td>Interviews</td>
<td>Yes</td>
</tr>
<tr>
<td>Do et al. (2010)</td>
<td>Analysis of the effects of time constraints on the costs and benefits of test case prioritization techniques</td>
<td>Experiment</td>
<td>Yes</td>
</tr>
<tr>
<td>Lohan et al. (2014)</td>
<td>Analysis of the effects of group cohesion and time pressure on decision-making quality in software development groups</td>
<td>Survey</td>
<td>Yes</td>
</tr>
<tr>
<td>Mäntylä and Itkonen (2013)</td>
<td>Analysis of the effect of time pressure on the effectiveness and efficiency of manual testing tasks</td>
<td>Experiment</td>
<td>Yes</td>
</tr>
<tr>
<td>Mäntylä et al. (2014)</td>
<td>Analysis of the effect of time pressure on the effectiveness and efficiency of test case development and requirements review</td>
<td>Experiment</td>
<td>Yes</td>
</tr>
<tr>
<td>Maruping et al. (2015)</td>
<td>Analysis of the effect of perceived time pressure on team performance and the moderating role of team temporal leadership in this context</td>
<td>Survey</td>
<td>Yes</td>
</tr>
<tr>
<td>Nan and Harter (2009)</td>
<td>Analysis of schedule pressure on software development cycle time and effort</td>
<td>Secondary data, survey</td>
<td>Yes</td>
</tr>
<tr>
<td>Pennington and Tuttle (2007)</td>
<td>Analysis of the effect of time pressure on software project risk assessment</td>
<td>Experiment</td>
<td>Yes</td>
</tr>
<tr>
<td>Shah et al. (2014)</td>
<td>Analysis of how vendor-side test engineers in a global software testing context accomplish their tasks under deadline pressure situations</td>
<td>Ethnography (interviews, observations, informal chats)</td>
<td>Yes</td>
</tr>
<tr>
<td>Sojer et al. (2014)</td>
<td>Analysis of the effect of time pressure on software developers’ attitude towards unethical programming behavior</td>
<td>Survey, interviews</td>
<td>Yes</td>
</tr>
<tr>
<td>Topi et al. (2005)</td>
<td>Analysis of the effects of time pressure on performance (i.e., speed, correctness) concerning the creation of database queries</td>
<td>Experiment</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 3. Overview of Analyzed Publications

Although not providing an explicit definition of time pressure, authors of several studies (Deak et al., 2016, Do et al., 2010, Lohan et al., 2014, Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Shah et al., 2014, Topi et al., 2005) seem to share the understanding provided in the definition by Maruping et al. (2015). Based on the work of Cooper et al. (2001) and Kelly and McGrath (1985), time pressure is typically defined as “the perception that there is a scarcity of time available to complete a task, or set of tasks, relative to the demands of the task(s) at hand” (Maruping et al., 2015, p. 1315). In contrast to this common understanding, authors of one study see time pressure as a discrepancy between the schedule estimated by the project team and the schedule negotiated with the client (Nan and Harter, 2009). Additionally, we found a reference to time pressure as “tendency of testing time to shrink from the original estimate until the actual testing execution period takes place” (Deak et al., 2016, p. 9). In other studies, time pressure is viewed as an objective condition (Austin, 2001, Pennington and Tuttle, 2007) rather than a subjective perception. Thereby,
the level of time pressure can be understood as the probability of a deadline being unrealistic (Austin, 2001) rather than the severity of time pressure. Despite not providing a clear definition, several researchers state their understanding of time pressure. Because time pressure is seen as an issue in every software project, it is only considered relevant if it is perceived as leading to ‘severe’ consequences (Sojer et al., 2014) or as “greater than average deadline pressure on the project” (Banker et al., 1987, p. 171). In only one study, time pressure is conceptualized as a two-dimensional construct, “where challenge time pressure produces a positive reaction and hindrance time pressure produces a negative reaction” (Lohan et al., 2014, p. 3).

In our sample of publications, time pressure is typically considered from the individual perspective (Austin, 2001, Banker et al., 1987, Deak et al., 2016, Do et al., 2010, Mäntylä et al., 2014, Pennington and Tuttle, 2007, Sojer et al., 2014, Topi et al., 2005). Other authors address the effect of time pressure on individuals as well, but emphasize the team context (Lohan et al., 2014, Mäntylä and Itkonen, 2013, Shah et al., 2014). In only two of the analyzed studies time pressure was treated as a team-level concept (Maruping et al., 2015, Nan and Harter, 2009).

Research Approach and Context

Here, we discuss our findings pertaining to the research approaches adopted and the types of data used in the studies included in our sample. Additionally, we examine the study context by analyzing the focus (e.g., software testing) and perspectives (e.g., developer) considered in each case.

With one exception (Austin, 2001), authors of all studies in the analyzed sample relied on empirical data to assess the role of time pressure in software projects (see Table 3). With the authors of five studies relying on experimental data (Do et al., 2010, Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Pennington and Tuttle, 2007, Topi et al., 2005) and four further cases based on surveys or secondary data (Banker et al., 1987, Lohan et al., 2014, Maruping et al., 2015, Nan and Harter, 2009, Sojer et al., 2014), a strong focus on quantitative research approaches is evident. Only in two studies (Deak et al., 2016, Shah et al., 2014) the authors made exclusive use of qualitative research methods and report findings based on the analysis of data gathered through interviews, observations, and informal chats.

In our sample, we identified two approaches to the assessment of the role of time pressure in software projects. First, several authors consider time pressure in software projects in general (Lohan et al., 2014, Maruping et al., 2015, Nan and Harter, 2009, Pennington and Tuttle, 2007). Second, others address time pressure in relation to particular activities, such as programming (Austin, 2001, Banker et al., 1987, Sojer et al., 2014, Topi et al., 2005) and software testing (Deak et al., 2016, Do et al., 2010, Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Shah et al., 2014).

Besides the studies focusing on the general team perspective (Maruping et al., 2015, Nan and Harter, 2009), the effects of time pressure have been assessed from a variety of perspectives, including both managerial roles such as project leaders (Banker et al., 1987) or risk auditors (Pennington and Tuttle, 2007), and technical roles such as software developers (Austin, 2001, Do et al., 2010, Lohan et al., 2014, Sojer et al., 2014, Topi et al., 2005) or software testers (Deak et al., 2016, Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Shah et al., 2014). Yet, despite examining time pressure from the perspective of different software development roles, in some of the experimental studies, students served as research subjects (Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Topi et al., 2005).

Effects of Time Pressure in Software Projects

When examining the effects of time pressure in software projects, authors of the studies included in our sample employed a variety of target variables, and thus reported different results, with diverse implications, as will be discussed below.

In line with the diversity of research approaches and contexts, the target variables used to assess the effects of time pressure also vary considerably across the analyzed studies. Some authors adopted indices referring to both general projects and individuals working on the projects. Indices in the former category pertain to productivity (Banker et al., 1987), task performance (Maruping et al., 2015, Topi et al., 2005), correctness (Topi et al., 2005), decision quality (Lohan et al., 2014), cycle time and effort (Nan and Harter, 2009), efficiency (Mäntylä et al., 2014), effectiveness (Mäntylä and Itkonen, 2013), and cost-effectiveness (Do et al., 2010). Shah et al. (2014) does not focus on the aforementioned indices, but rather on individuals’ general perceptions of time pressure in software projects. Concerning the individual perspective, target variables include information overload (Pennington and Tuttle, 2007) and developers’ behavior related to the taking of shortcuts (i.e., software imperfections that developers deliberately
induce to save time and meet a tight deadline; see Austin, 2001), motivation (Deak et al., 2016), and unethical reuse of code (Sojer et al., 2014). Although these variables relate to individuals instead of projects in their entirety, they can ultimately harm company reputation.

The findings related to time pressure’s role in software projects are as diverse as suggested by previous studies in which time pressure’s effects in other settings were considered (see Time Pressure in Software Projects). In some studies, either a positive or a negative effect of time pressure on project outcomes is reported. On the one hand, several authors observed a positive effect of time pressure on productivity in the execution of software projects (Banker et al., 1987), effectiveness and efficiency of software testing (Mäntylä and Itkonen, 2013, Mäntylä et al., 2014), decision quality (Lohan et al., 2014), and software quality (Austin, 2001). On the other hand, some researchers found that time pressure leads to unethical code reuse (Sojer et al., 2014), information overload (Pennington and Tuttle, 2007), decreased motivation (Deak et al., 2016), and fewer correctly performed database queries (Topi et al., 2005). In addition to the aforementioned research streams in which time pressure was found to be favorable or detrimental to performance, authors of two studies report the inverted U-shape relationship between time pressure and performance in software projects (Maruping et al., 2015, Nan and Harter, 2009). While Maruping et al. (2015, p. 1323) state that their findings do “not reflect the full inverted U-shape relationship, the pattern provides partial support for Hypothesis 1 ['Time pressure will have an inverted-U relationship with team processes.']”, the results of Nan and Harter (2009) do not support a significant U-shape relationship between schedule pressure and cycle time or development effort.

Due to the incongruence in the reported results, it is challenging to derive practical implications from previous research concerning time pressure in software projects. As noted by Mäntylä et al. (2014), practitioners typically associate time pressure with negative outcomes. Examples for such associations include time pressure as a reason for lower software quality, burnout, and reduced job satisfaction. Findings yielded by several of the analyzed studies corroborate this negative association (Pennington and Tuttle, 2007, Sojer et al., 2014, Topi et al., 2005). As a consequence of this view, their authors recommend that time pressure be better managed, which might be a challenging endeavor because the often unrealistically low effort estimates (Basten and Sunyaev, 2014) make time pressure a common condition in software projects. However, the answer to the question whether time pressure inhibits or contributes to success of software projects might depend on the degree of time pressure. In contrast to findings that indicate time pressure’s negative effect on project outcomes, some authors claim that proper levels of time pressure might have beneficial effects on project outcomes (Nan and Harter, 2009). For instance, the link between time pressure and unethical code reuse has only been revealed in cases of severe time pressure. Research concerning software testing also shows that time pressure can increase both effectiveness and efficiency (Mäntylä and Itkonen, 2013, Mäntylä et al., 2014). Maruping et al. (2015, p. 1328) thus indicate that “some degree of time pressure is beneficial for motivating teams to engage in team processes that facilitate performance”. This suggestion is in line with the view that time pressure should be managed so that it is perceived as challenging rather than hindering (Lohan et al., 2014). Additionally, time pressure is suggested as a short-time measure only. A long-term strategy based on time pressure is likely to affect staff morale and result in increased turnover (Banker et al., 1987). Mäntylä and Itkonen (2013, p. 1000) conclude: “Use time pressure, but not constantly, as it may backfire in the form of a high staff turnover rate”. Finally, although Austin’s (2001) investigation does not extend to the severity of time pressure, his game theory model suggests that continuously high time pressure results in a state in which the shortage of time is so pervasive that reporting delays becomes destigmatized. Consequently, developers tend not to take shortcuts, which would ultimately reduce software quality, thereby mitigating the adverse effects of time pressure.

RESEARCH AGENDA

In this section, we discuss our findings based on which we develop an agenda for future research concerning the role of time pressure in software projects. We present five themes that authors of future works should consider: methodological pluralism, conceptualization, contemporary development approaches, the role of context, and empirical validation.

Methodological Pluralism

In the set of analyzed publications, we found a strong focus on quantitative research. Most authors relied on experiments or surveys for collecting data needed to examine the role of time pressure in software projects. While those studies provide helpful insights into effects of time pressure, the resulting assessment of time pressure is also restricted to the setting chosen in the study. Few studies have addressed time pressure in software projects from a qualitative perspective. The studies by Deak et al. (2016) and Shah et al. (2014) are the only two in our sample in
which a purely qualitative research approach (an ethnographic-informed study based on interviews, observations, and informal chats) was adopted.

A complex phenomenon such as time pressure, from our perspective, requires methodological pluralism to be understood in its entirety. Research conducted to date has revealed a positive, a negative, as well as an inverted U-shape relationship between time pressure and the outcomes of software projects. Qualitative research, such as qualitative field studies and case study research, is likely to contribute to a better understanding of time pressure in the context of software projects by providing in-depth insights from experts and triangulating data from various perspectives. For example, longitudinal case studies can help assess the effects of time pressure in the long term.

**Conceptualization**

In most of the studies included in our literature review, the authors do not provide a definition of time pressure. This is likely to be problematic because a generally accepted definition of the concept is lacking (Hwang, 1994). Without taking the definition and understanding of time pressure into account, it can be misleading to derive general implications from extant research. Findings yielded by studies based on different understandings of time pressure should not be compared or combined without reconciling their respective conceptualizations and operationalizations of the core concept.

Considering two extremes, whereby time pressure is defined as the perception of time available to complete a task being scarce relative to the demands of that task (Maruping et al., 2015) and time pressure is treated as the probability of task deadlines being unrealistic (Austin, 2001), we encourage authors of future studies to clearly conceptualize time pressure. Depending on the conceptualization, it might be important to emphasize whether the concept is measured objectively, or based on the perception of project stakeholders. In the latter case, the stakeholders considered should be explicitly named. In conceptualizing time pressure, experience and skills of stakeholders are also decisive, since they influence the perception of the severity of time pressure (e.g., a person familiar with a task is less likely to perceive allocated time as inadequate for its completion compared to someone without the requisite experience).

**Contemporary Development Approaches**

Authors of several publications in our sample examined a specific phase of software projects, such as software testing (Deak et al., 2016, Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Shah et al., 2014). Other authors consider software projects in their entirety (Austin, 2001, Lohan et al., 2014, Maruping et al., 2015). In cases without explicit descriptions of the context, research seems to focus on software projects following traditional development approaches. This assumption is based on several considerations. First, some of those cases pertain to older publications (e.g., Austin, 2001, Banker et al., 1987), which are unlikely to refer to agile development approaches, considering that their popularity surged at the beginning of the 21st Century. Second, several works have an explicit focus on a single development phase (e.g., Mäntylä and Itkonen, 2013, Mäntylä et al., 2014, Shah et al., 2014), which is uncommon for agile software projects. Finally, the fact that the development approach is not specified points towards traditional approaches, because authors with a focus on contemporary ones would be likely to mention these in their research approach. While Deak et al. (2016) refer to both sequential and agile development, the study conducted by Malgonde et al. (2014), which we excluded from our sample due to its status as a research-in-progress paper, is the only one with an explicit focus on agile software development. In their research, Malgonde et al. (2014) aim to establish whether agile software development approaches can be adopted to effectively handle time pressure. Based on Extreme Programming, the authors make a first step to shift attention to the phenomenon of time pressure in agile software projects.

Yet, research concerning the role of time pressure in agile software projects is limited. Since time pressure is a problem likely to result from inadequate project management, Scrum as an agile project management approach that is applicable to software projects (see *Software Projects*) is a fruitful avenue for future research. In future studies, it would be thus beneficial to consider a juxtaposition of sequential and agile development approaches, as this would allow the authors to investigate whether contemporary, agile development approaches are more suitable for mitigating time pressure in software projects due to their shorter cycle times.
The Role of Context

Findings yielded by the extant research examining the impact of time pressure software project outcomes are inconclusive (see Effects of Time Pressure in Software Projects). Considering the various contexts that have been used to study time pressure in software development (e.g., different project phases) and the diverse results, we suggest being more mindful of the context in which the effects of time pressure are analyzed. Since previous research has shown that measuring project performance depends on the project context (Pankratz and Basten, 2015) and time pressure is likely to impact project performance, future research should identify and consider theoretical perspectives that fit the respective context. Researchers should, for instance, consider focusing on the perspective from which time pressure is assessed (e.g., developers) and how time pressure affects those individuals in their work routines.

Empirical Validation

When evaluating the publications in our sample, we noted that Austin (2001) is the only author that does not rely on empirical data. This study is also unique for two other reasons. First, time pressure is considered as a probability of deadlines being unrealistic. Second, Austin (2001) uses game theory to explore the relationship between time pressure and software quality and develops the counter-intuitive hypothesis that increasing time pressure leads to better software quality. According to the proposed model, being continuously subjected to high time pressure results in a state in which the shortage of time is so ever-present that reporting delays is destigmatized, ultimately leading to less shortcut-taking and thus higher software quality. Despite manifold references to Austin’s (2001) work in the information systems literature (e.g., Asdemir, Kumar and Jacob, 2012, Fisher, Chengalur-Smith and Ballou, 2003, Levina, 2005, Mahaney and Lederer, 2003, Mahaney and Lederer, 2006, Nan and Harter, 2009, Shah et al., 2014, Shao, Yin and Chen, 2014, Wang, Ju, Jiang and Klein, 2008), the model has not been empirically tested yet. Accordingly, we suggest that authors of future studies identify suitable means for empirically testing Austin’s (2001) counter-intuitive hypothesis.

Besides the evaluation of the model proposed by Austin (2001), we encourage researchers to replicate the empirical studies included in our review. Considering the variety of conceptualizations and research contexts, each of the analyzed studies is unique. Replication of these studies would thus help strengthen the confidence in the reported findings. This call is in line with the increased awareness of the value of replication studies in the information systems discipline (Dennis and Valacich, 2014, Niederman and March, 2015).

CONCLUSION

In this work, we aimed to advance research on time pressure’s role in software projects by providing a synthesis of pertinent studies and developing a research agenda that paves the way for future studies in this domain. Our literature review reveals that authors of most studies rely on quantitative research approaches and aim to analyze the effect of time pressure on outcomes of projects applying sequential development approaches. Future research should thus focus on contemporary development approaches that provide an alternative project management strategies, such as Scrum. Thereby, attention should be paid to the conceptualization of time pressure and the study context. The lack of a generally accepted definition of time pressure and the diverse contexts studied might account for the inconsistent picture concerning time pressure’s role in software projects that has emerged from this investigation.

Our study is limited by the depth of the analysis provided in this paper and the number of publications analyzed. Concerning the former, a systematic mapping study (Kitchenham, 2007) of the identified studies will help assess the strength of evidence that previous research provides in support of the reported time pressure’s effects on software development outcomes. We omitted such an assessment because our primary purpose was to develop a research agenda. Concerning the latter, our review covers the manifold outlets considered in the AIS journal ranking and is thus likely to provide a comprehensive perspective on time pressure’s role in software projects. Additionally, we applied both backward and forward searches to decrease the likelihood of overlooking relevant studies. While we identified several other studies in which time pressure is discussed in the context of software projects, these were excluded, as their authors did not assess the role of time pressure.

Even though our primary objective was the development of a research agenda, the aggregation of the analyzed studies also has some important practical implications. First, the reported findings indicate that severe levels of time pressure should be avoided. Otherwise, members of software projects might adapt their behavior in ways that negatively affect project success. Second, moderate levels of time pressure can be used to increase project success by reducing slack,
motivating higher employee performance, and ultimately increasing productivity. Finally, time pressure should be used as a short-term measure only, as working for prolonged periods under excessive time pressure reduces staff morale and increases turnover.

REFERENCES


Team Performance in Agile Software Development Projects: The Effects of Requirements Changes, Time Pressure, Team Diversity, and Conflict

Phil Diegmann
University of Cologne
diegmann@wiso.uni-koeln.de

Christoph Rosenkranz
University of Cologne
rosenkranz@wiso.uni-koeln.de

ABSTRACT

Information system development has traditionally been accompanied by changing requirements throughout projects. While recent research clarifies how time pressure affects team performance, the interplay between time pressure, requirements changes, and various other important factors such as team diversity is unclear. In this paper, we evaluate a novel and unified model based on extant research explaining the interactions of requirements changes, time pressure, temporal leadership, team diversity, and conflict as well as their effect on team performance. Further, we differentiate between the overall team performance and the team’s ability to respond both efficiently and extensively to requirements changes. The proposed model helps in composing and steering teams to increase their resilience towards requirements changes and time pressure. A first evaluation is based on a quantitative field study in multiple agile software development projects of student teams.

Keywords
Team Performance, Agile Software Development, Requirements Changes, Time Pressure, Diversity, Conflict

INTRODUCTION

Agile methods for information system development (ISD) are increasingly popular in industry (Conboy 2009; Dybå and Dingsøyr 2008; Fitzgerald et al. 2006; Lee and Xia 2010; Williams 2012) and see wide-spread adoption (VersionOne 2017). Agile ISD methods, however, are no guarantee that a team will be successful (Fraser and Mancl 2008), and our understanding of what affects team performance and of the mechanisms of action in agile ISD (henceforth: AISD) teams is lacking (Lee and Xia 2010; Persson et al. 2012; Sarker et al. 2009). Team-level research in AISD, however, is scarce (Lee and Xia 2010), although AISD is mostly conducted in teams and is quintessentially a team effort (Siau et al. 2010). Moreover, results are inconsistent. Some studies suggest that AISD methods work best for highly cohesive and similar (i.e., non-diverse) teams (Cao et al. 2009; Fruhling and de Vreede 2006), and that cohesiveness could be the main reason for successful ISD. Others find that diversity amplifies creativity and communication, and therefore contributes to the success of AISD methods (Bear and Woolley 2011; Lee and Xia 2010; Phillips et al. 2006). However, some studies identified an overhead in communication due to diversity in teams, which hinders performance (Ely and Thomas 2001; Leonard et al. 2004; MacMillan et al. 2004). Similar to diversity, time pressure has been found to both hinder and facilitate team performance, depending on the temporal leadership, which helps to mitigate negative effects of time pressure (Maruping et al. 2015).

Research on teams and team performance therefore has identified a need to move beyond the simple diversity-affects-performance model (Van Der Vegt and Bunderson 2005, p. 542), with calls for more empirical research on how different factors affects team performance (henceforth: performance) in AISD (Lee and Xia 2010), and on team-level effects in AISD (Conboy 2009; Mangalaraj et al. 2009; McAvoy and Butler 2009; McAvoy et al. 2013).

In this study, we aim to (1) identify the effects of diversity in AISD teams, (2) and to consolidate previous findings on team performance (especially regarding time pressure and temporal leadership) into a unified and richer understanding of team performance in AISD. Explaining team performance in a unified, consolidated, and parsimonious model will ultimately help to understand AISD better, and to reduce the number of failed AISD projects. Therefore, we enrich our understanding of team performance in AISD by including and adapting findings of team- and organizational-behavior research as well as established findings from the information systems (IS) research community.
We tested our model with a sample from eight student teams comprising 33 students and engaged in AISD projects, using partial-least square (PLS) structural equation modeling. We found that team performance is affected by requirements changes, temporal leadership, and team diversity. Time pressure has been found to increase affective conflict, which in turn decreased team performance. Further we found that response efficiency precedes and improves response extensiveness, which in turn showed to be beneficial for team performance.

Our results corroborate our unified model and indicate that in an enriched and extended model of team performance interdependencies and effects can manifest differently from more contained models as found in extant research. We therefore expand previous research on team performance and AISD by taking the bigger picture of AISD into account.

The remainder of this paper is structured as follows. First, we lay out related work and establish the theoretical background. Second, we argue for our hypotheses and lay out our research model. Third, we present our research design and explain the details of data analyses. These results are then discussed. Finally, we discuss the findings and limitations of this study, give an outlook to future research and give a conclusion of this paper.

RELATED WORK

Information Systems Development & Agile Approaches

IS are often developed in the form of projects (Hirschheim et al. 1995, p. 33), with many involved stakeholders and project team members (Chae and Poole 2005). The nature of ISD is in many aspects intangible (Cule et al. 2000). As a result, many problems of ISD projects are not so much technological as sociological in nature (DeMarco and Lister 1987, p. 4). Coordination and communication between various stakeholders are necessary for successful implementation (Corvera Charaf et al. 2013; Gallivan and Keil 2003; Ko et al. 2005), and creating a shared understanding between involved stakeholders is deemed to be a major driver for ISD success (Bittner and Leimeister 2014; Gallivan and Keil 2003; Rosenkranz et al. 2013; Rosenkranz et al. 2014; Tan 1994).

In practice, methods for developing IS range from sequential approaches (Royce 1970) to more cyclic, iterative approaches (Boehm 1988). The contemporary state-of-the-art AISD methods (Cao et al. 2009; Vidgen and Wang 2009), which are increasingly adopted in industry (VersionOne 2017), trade strict control for more flexibility and autonomy within the team, the overall development process is not planned upfront, and progress is made in small iterative phases, while encouraging change and constant feedback (Cockburn and Highsmith 2001; Highsmith and Cockburn 2001). Planning becomes a permanent task, and team leadership is established via collaboration (Dybå and Dingsøyr 2008; Dybå and Dingsøyr 2009).

While the team thus is highlighted as crucial to AISD in practice, extant research in the field of AISD methods has investigated mainly specific, individual, or organizational phenomena, such as the use and effects of specific agile practices (Balijepally et al. 2009; Holmqvist and Pessi 2006; Maruping et al. 2009b), and effects regarding whole projects or organizations, such as the adoption of AISD methods (Cao et al. 2009; Heeager 2012; Hong et al. 2011; Kotlarsky 2007; Mangalaraj et al. 2009).

As research thus covers the individual and organization-wide level of effects on AISD, team-level effects are covered less so. Likewise, team research in organizational behavior has included technology as an influencing factor of team work (e.g., Kozlowski and Ilgen 2006), but specific features of ISD have not been observed. Research found that cohesive teams are the optimal base for applying agile practices (Cao et al. 2009; Fruhling and de Vreede 2006), while other studies suggest that diversity – the differences among team members regarding visible (i.e., surface-level; e.g., ethnicity, age) and invisible (i.e., deep-level; e.g., experience, education) characteristics (Bear and Woolley 2011; Lee and Xia 2010; Phillips et al. 2006) – amplifies creativity and problem-solving ability (Bear and Woolley 2011; Lee and Xia 2010; Phillips et al. 2006) and therefore might provide benefits for ISD. These inconsistencies are especially important for AISD, as AISD teams rely heavily on efficiency (to respond quickly to requirements changes; Conboy 2009) and problem solving ability (to complete complex, non-routine tasks; Lee and Xia 2010).

AISD explicitly acknowledges the importance of being able to respond to requirements changes and even embrace change and an ever-changing environment (Beck et al. 2001). Each change imposes difficulties for the team. Therefore, AISD teams have to have the capacity to recover quickly from these changes and resulting difficulties. Extant research proposed this ability to consist of response efficiency (i.e., responding to and implementing of requirements changes with minimal time, cost, personnel, and resources) and response extensiveness (i.e., the
proportion of requirements changes that a team responds to and implements) – both have been found to be important predictors of overall performance (Lee and Xia 2010).

As AISD embraces requirements changes and aims at responding efficiently and extensively, changing requirements are leading to additional work. Therefore, AISD teams are exposed to time pressure.

**Time Pressure & Temporal Leadership**

Time pressure is defined as the perceived scarcity of time to complete a task (Cooper et al. 2001). It is important to differentiate time pressure from performance pressure (the shared accountability for outcomes, high scrutiny of work, and significant consequences of performance outcomes; Gardner et al. 2012), as well as from urgency (a concern for time and a feeling of being continuously hurried; Mohammed and Nadkarni 2011). Extant research (Maruping et al. 2015) found that perceived time pressure (in the following simply named “time pressure”) negatively affects team processes. These processes in turn contribute to performance. With increased time pressure and therefore weakened team processes, performance is bound to decrease.

Actions which structure, organize, or manage a team and its tasks are conceptualized as temporal leadership (Maruping et al. 2015). This concept has been found to moderate the effect of time pressure, meaning that high temporal leadership offsets negative effects on processes and ultimately performance (Maruping et al. 2015). Temporal leadership therefore plays an important role in AISD, due to its change-embracing nature.

Changing requirements lead to additional work and new tasks. Typically, deadlines cannot be moved – which leads to more work per time. While requirements changes are therefore affecting time pressure and project failure being partly attributed to changing requirements (Zowghi and Nurmuliani 2002), recent research has not yet integrated requirements changes into the interactions of time pressure, temporal leadership, and performance.

Similar to requirements changes, findings in organizational research have not yet been fully integrated in ISD research, although diversity in AISD teams becomes increasingly important, due to outsourcing and globally distributed teams.

**Diversity & Conflict**

Behavioral research on team work has focused mainly on outcomes of performance before shifting from input-process-output models to cyclic input-mediation-output-input models (Ilgen et al. 2005). A notion of teams as complex, context-sensitive, and evolving systems has emerged (Ilgen et al. 2005; Kozlowski and Bell 2003).

As ISD projects are becoming more distributed and diverse (e.g., Persson et al. 2012; Ramesh et al. 2012; Sarker et al. 2009; Sarker and Sarker 2009), research on AISD has started adapting diversity concepts, and calling for a better understanding of effects of diversity in ISD (Lee and Xia 2010). Extant research applied theories of organizational psychology while being focused on IT use than on ISD (e.g., Gorecki et al. 2008; Nan 2011; Wang and Hahn 2015).

In organizational psychology, diversity has emerged as an important predictor of performance and research found contradictions (del Carmen Triana et al. 2014; Hülshheger et al. 2009; Joshi and Roh 2009; Milliken and Martins 1996; Phillips et al. 2006; Post 2012; Van Der Vegt and Bunderson 2005). Some studies find a positive relation between diversity and performance (see Bear and Woolley 2011; Phillips et al. 2006; Van Der Vegt and Bunderson 2005), but outlined a dependency on specific contextual circumstances, such as the competitive threat-level (del Carmen Triana et al. 2014), team identification, and climate (Van Der Vegt and Bunderson 2005). Team identification and climate have been found to play an important role in generating positive effects from diversity (Van Der Vegt and Bunderson 2005). Studies which identified a negative effect describe an overhead of communication and a risk for conflict (Ely and Thomas 2001; Leonard et al. 2004; MacMillan et al. 2004).

Scholars differentiate between deep-level (DLD; i.e. education, experiences) and surface-level diversity (SLD; i.e. ethnicity, age) (Aggarwal and Woolley 2013; Hülshheger et al. 2009; Phillips et al. 2006). These two types act differently: while SLD highlights dissimilarities and encourages sharing of unique information (Phillips et al. 2006), DLD might lead to harmful conflict (Jehn et al. 1999) or facilitate performance by providing different educational backgrounds and skillsets (Joshi and Roh 2009).
Furthermore, diversity can be grouped (Harrison and Klein 2007) into separation (differences in opinions etc.), disparity (differences in socially valued assets), and variety (differences in knowledge etc.). In the scope of this research, we are most interested in variety, as differences in knowledge can help overcome crises.

Diversity entails a risk of increased conflict and a need for increased communication among team members. Research distinguishes between affective conflict (i.e., conflict on an emotional or personal level that is not task-related; also named relationship conflict) versus task conflict (i.e., disagreements about ideas and strategies regarding the current task) (Jehn et al. 2008). While a medium level task conflict has been found to be beneficial to non-routine tasks, affective conflict has been found to negatively affect performance (Jehn et al. 2008).

In sum, no integrated model exists that explains the relationships between factors such as diversity, temporal leadership, conflict, requirements changes, and time pressure. This becomes especially important if the inconsistent findings of effects of diversity are taken into account. A holistic model integrating the findings of organizational research into extant ISD research helps in explaining interdependencies and interactions of existing models.

**RESEARCH MODEL & HYPOTHESES**

Based on the literature laid out above, we argue for a novel research model that integrates these streams for AISD (see Figure 1). Table 1 provides detailed definitions for each construct.
Due to the change-embracing nature of AISD (Beck et al. 2001), we focus on requirements changes first. Requirements changes are conceptualized as a change in the customer’s understanding of their own needs over the course of the project (Maruping et al. 2009a). Likewise, time pressure is defined as the perceived scarcity of available time to finish a task (Maruping et al. 2015). As requirements changes have been found to increase time pressure due to increased work amounts (Maruping et al. 2015), we propose:

**H1:** Increased requirements changes increase time pressure.

With increased time pressure, confusion might arise over who is responsible for which task, due to limited resources (Maruping et al. 2015). Furthermore, this results in limited time to resolve existing conflicts or inhibits to blights upcoming conflicts. This further increases frustration and in turn conflict (Maruping et al. 2015) – more specifically, affective conflict, that is, conflict on an emotional or personal level, which is not directly task-related (Jehn et al. 2008). Following this argument, we state:

**H2:** More time pressure leads to more affective conflict.

In line with the argumentation of Maruping et al. (2015), we propose a moderation of temporal leadership on the effect of time pressure on affective conflict. Temporal leadership is defined as “leader behaviors that aid in structuring, coordinating, and managing the pacing of task accomplishment within the team” (Mohammed and Nadkarni 2011, p. 492); temporal leadership plays an important role to utilize time pressure as a motivator. As we focus on affective conflict (i.e., the negative effects of time pressure due to increased confusion about who should complete what tasks (Maruping et al. 2015)), we propose that strong temporal leadership leads to less time-pressure-related conflicts. Therefore, we propose:

**H3:** Higher levels of temporal leadership decrease the effect of time pressure on affective conflict.

Diversity has been found to have both positive (Bear and Woolley 2011; Lee and Xia 2010; Phillips et al. 2006) and negative effects (Cao et al. 2009; Fruhling and de Vreede 2006) on performance – the fulfillment of requirements, functionality, and quality. This depends on the level of diversity and the type of resulting conflict (Jehn et al. 1999;
Joshi and Roh 2009; Phillips et al. 2006). As we focus on diversity in skillsets and experiences (Joshi and Roh 2009) to tackle shocks resulting from requirements changes, we are especially interested in deep-level diversity, which has been found to increase the need for communication and entailing a risk of misunderstanding and conflicts (Jehn et al. 1999). Therefore, we propose:

\( H_4 \): Higher levels of diversity lead to more affective conflict.

As extant research found, affective conflict is hindering performance (e.g., by requiring additional time to resolve conflicts; Jehn et al. 2008) which is why we state:

\( H_5 \): Increased affective conflict decreases performance.

Coming back to the top of our model, we further expect requirements changes to increase response efficiency. The more requirements changes are brought forward by a customer, the more potential a team has to respond efficiently. Therefore, we posit:

\( H_6 \): Higher levels of requirements changes lead to more response efficiency.

Extant research found that with increasing time pressure, developers engage in less productive, inefficient tasks, constraining their ability to efficiently complete their task (Durham et al. 2000; Perlow 1999). If time pressure rises, the ability to turn requirements changes into response efficiency (see \( H_5 \)) is being limited. If \( H_1 \) is supported, this assembles a control-loop: increased requirements changes increase the potential of response efficiency which in turn is limited if the increased requirements changes raise the time pressure. We expect this to ultimately prevent teams from responding efficiently for high time pressure.

\( H_7 \): More time pressure decreases the enabling effect of requirements changes on response efficiency.

Regarding temporal leadership (i.e., “leader behaviors that aid in structuring, coordinating, and managing the pacing of task accomplishment within the team” (Mohammed and Nadkarni 2011, p. 492)), we expect temporal leadership to increase response efficiency. Contrary to extant literature (Maruping et al. 2015), we do not posit time pressure to have a direct effect on response efficiency (and therefore temporal leadership not having a moderating effect on response efficiency). Instead of replacing it, we modify and transform the previous model. \( H_3 \) resembles the moderating and mitigating effect of temporal leadership on the negative effects of time pressure. We further argue that temporal leadership increases response efficiency due to the structuring, coordinating, and managing (i.e., optimizing) actions associated with strong temporal leadership. Therefore, we propose:

\( H_8 \): Higher levels of temporal leadership lead to increased response efficiency.

With the ability to respond efficiently to requirements changes, we expect teams to be able to also respond more extensively. While extant research found a negative effect of response extensiveness on response efficiency (Lee and Xia 2010), an inverted relationship so far has neither been supported nor dismissed. We argue that with high response efficiency, resulting from strong temporal leadership, a team is able to respond more extensive than otherwise. We acknowledge that an overly extensive response can have adverse effects on response efficiency (i.e. due to an overwhelming effect (Lee and Xia 2010)), but we argue that this situation would not have happened if a strong temporal leadership was in place (e.g., due to a better structuring and organization of tasks). Therefore, we state:

\( H_9 \): Increased response efficiency increases response extensiveness.

As a team is able to respond extensively to requirements changes, performance is bound to increase as well, as this results in more software functionality being implemented as initially needed by the client (Lee and Xia 2010):

\( H_{10} \): Increased response extensiveness increases performance.

As our sample consists of students of a graded course, we expect an increased performance to lead to better grades:

\( H_{11} \): Increased performance leads to better grading.
RESEARCH DESIGN

Survey Design & Data Collection

As our research model is based on extant literature and all constructs were already used by previous literature, we were able to reuse parts of existing measurement instruments.

In accordance with the quantitative nature of our study we used a questionnaire to gather data. Over the course of an undergraduate class in which students had to develop software for practitioners following AISD principles (Scrum), we provided two different surveys at two different times. First and at the very beginning of an iteration, we asked for general demographics and static information and control variables, such as the experience developing software. Second and just after finishing the iteration, we asked the participants for all outcome-related constructs. Third, we measured the grading of the outcome by the teaching assistants and the industry partner at the end of the project. With this approach, we hope to get an as unbiased result as possible, as we gather static information (e.g., diversity) without distortion from current, quite possibly stressed, emotions, while still gather time-pressure-sensitive data during the most stressful part of AISD (i.e., sprint completion).

All participants were granted anonymity and did not receive any rewards for participation. The grades were added to our dataset by an otherwise not involved researcher without access to any identifying properties aside from the student-identification number.

Measurement Scales

Table 2 lists all constructs used in our research model as well as the measurement items for each construct. To ensure a common understanding of all items, we applied the back translation method (Brislin 1970), as all participants were more proficient in German than in English.

<table>
<thead>
<tr>
<th>Construct</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RC) Requirements Changes (Maruping et al. 2009a)</td>
<td>Requirements fluctuated quite a bit in early phases of the project</td>
</tr>
<tr>
<td></td>
<td>Requirements fluctuated quite a bit in later phases of the project</td>
</tr>
<tr>
<td></td>
<td>Requirements identified at the beginning of the project were quite different from those toward the end</td>
</tr>
<tr>
<td>(TP) Time Pressure (Maruping et al. 2015)</td>
<td>We are often under a lot of pressure to complete our tasks on time</td>
</tr>
<tr>
<td></td>
<td>We are not afforded much time to complete our tasks</td>
</tr>
<tr>
<td></td>
<td>The amount of time provided to complete our tasks is short</td>
</tr>
<tr>
<td></td>
<td>Task durations are often short</td>
</tr>
<tr>
<td>(TL) Temporal Leadership (Maruping et al. 2015)</td>
<td>To what extent does the team leader remind members of important deadlines?</td>
</tr>
<tr>
<td></td>
<td>To what extent does the team leader prioritize tasks and allocate time to each task?</td>
</tr>
<tr>
<td></td>
<td>To what extent does the team leader prepare and build in time for contingencies, problems, and emerging issues?</td>
</tr>
<tr>
<td></td>
<td>To what extent does the team leader pace the team so that work is finished on time?</td>
</tr>
<tr>
<td></td>
<td>To what extent does the team leader urge members to finish sub-tasks on time?</td>
</tr>
<tr>
<td></td>
<td>To what extent does the team leader set milestones to measure progress on the project?</td>
</tr>
<tr>
<td></td>
<td>To what extent is the team leader effective in coordinating the team to meet customer deadlines?</td>
</tr>
<tr>
<td>(RE) Response Efficiency (Lee and Xia 2010)</td>
<td>How much additional effort was required by the team to incorporate the following changes? (Effort includes time, cost, personnel, and resources)</td>
</tr>
<tr>
<td></td>
<td>System Scope</td>
</tr>
<tr>
<td></td>
<td>System Input Data</td>
</tr>
<tr>
<td></td>
<td>System Output Data</td>
</tr>
<tr>
<td></td>
<td>Business Rules &amp; Processes</td>
</tr>
<tr>
<td></td>
<td>Data Structure</td>
</tr>
<tr>
<td></td>
<td>User Interface</td>
</tr>
<tr>
<td>(RX) Response Extensiveness (Lee and Xia 2010)</td>
<td>How much additional effort was required by the team to incorporate the following changes? (Effort includes time, cost, personnel, and resources)</td>
</tr>
<tr>
<td></td>
<td>System Scope</td>
</tr>
</tbody>
</table>
Construct | Item
--- | ---
| **System Input Data** | The members of the team were from different areas of expertise
| **System Output Data** | The members of the team had skills that complemented each other
| **Business Rules & Processes** | The members of the team had a variety of different experiences
| **Data Structure** | The members of the team varied in functional backgrounds
| **User Interface** | (TD)

**Team Diversity**
(Lei and Xia 2010)

| **How much have you managed friction among members of your project team?** | (AC)

**Affective Conflict**
(Maruping et al. 2015)

| **How much have you managed personality conflicts between team members during the project?** | (PE)

**Team Performance**
(Lee and Xia 2010; Maruping et al. 2015)

| **How much have you dealt with tension among members of your project team?** | (GR)

**Grading**

| **How much have you managed emotional conflict among members of your project team?** | The client perceives that the system meets intended functional requirements
| **The overall quality of the developed system is high** | The software delivered by the project achieved its functional goals
| **The software delivered by the project met end-user requirements** | The capabilities of the software fit end-user needs
| **The software met technical requirements** | The final grade (in points) for this sprint. Based on functionality implemented, bugs fixed, and code quality, as rated by the lecturer and practice partner.

**Table 2. Constructs & corresponding items.**

**DATA ANALYSIS & RESULTS**

We gathered data from 33 voluntarily partaking students. Table 3 describes the sample characteristics. The group to which the students were assigned did only correlate with one item – the functional background diversity (#4 of the diversity construct; r(31) = -.380, p < .05) –, indicating variations in perceived functional backgrounds among groups.

**Participants (n = 33)**

| Gender | Female (N = 5; 15.2%) | Male (N = 28; 84.8%) | Other (N = 0; 0%) |
| Age | 22.41 years average (range: 20-28) |
| Experience in working agile | No previous experience (N = 21; 63.6%) | One year (N = 7; 21.2%) | Two or more years (N = 4; 12.2%) | No answer (N = 1; 3.0%) |
| Group Sizes | Four to six students per Group (M = 5.00, SD = .5) |

**Table 3. Sample Description**

In the following we test our model using partial least squares (PLS) structural equation modeling. PLS is better suited for exploratory research and predict key target constructs using latent variables, and for complex structural models than a covariance-based approach (Hair et al. 2011). Our results were calculated with SmartPLS 3.2.6 (Ringle et al. 2015).

Our measurement model contains reflective and formative constructs. As the grading is a single-item and – in this context – less subjective construct compared to other success measures such as self-report performance, we do not check this construct for reliability or validity. We first checked the reflective constructs (i.e., all but response extensiveness and response efficiency) for reliability and validity. First, internal consistency was evaluated using Cronbach’s alpha and composite reliability which need to exceed .70 (Nunnally 1978; Werts et al. 1974). The constructs fulfill both criteria (see Table 5). Second, indicators are considered reliable if the associated latent construct
explains more than half of the indicator’s variance (Henseler et al. 2009); our model passes this criterion (see Table 6). Therefore, we assume indicator reliability.

Third, Table 5 shows that the composite reliabilities of all constructs exceed the required minimum of .80 and that the AVE values of all constructs exceed the threshold of .50. Additionally, all item loadings exceed the threshold of .70. Thus, convergent validity conditions are met (Fornell and Larcker 1981).

Fourth, the square root of each construct’s AVE needs to exceed the correlations with the other constructs to indicate discriminant validity (Fornell and Larcker 1981) which is the case for all of our constructs (see Table 4). Moreover, we examined the factor loadings of each indicator (Fornell-Larcker criterion). Each indicator needs to load higher on the associated construct compared to all other factors (Chin 1998) which is the case for our data (see Table 6). In addition to the Fornell-Larcker criterion, the Heterotrait-Monotrait (HTMT) ratio of correlations is suggested as a new criterion to assess discriminant validity (Henseler et al. 2015). The highest HTMT value for our model of .78 is below a conservative threshold of .85 (Henseler et al. 2015). Combining the results, we posit discriminant validity (Voorhees et al. 2015).

<table>
<thead>
<tr>
<th>RC</th>
<th>TP</th>
<th>TL</th>
<th>TD</th>
<th>AC</th>
<th>PE</th>
<th>GR</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>.84</td>
<td>.35</td>
<td>-16</td>
<td>.33</td>
<td>-16</td>
<td>.85</td>
</tr>
<tr>
<td>TP</td>
<td>.86</td>
<td>.90</td>
<td>.90</td>
<td>.74</td>
<td>.35</td>
<td>.85</td>
</tr>
<tr>
<td>TL</td>
<td>.35</td>
<td>.88</td>
<td>.83</td>
<td>.76</td>
<td>.35</td>
<td>.03</td>
</tr>
<tr>
<td>AC</td>
<td>.30</td>
<td>.93</td>
<td>.33</td>
<td>.93</td>
<td>.93</td>
<td>.93</td>
</tr>
<tr>
<td>PE</td>
<td>.03</td>
<td>.04</td>
<td>.32</td>
<td>.10</td>
<td>.04</td>
<td>.10</td>
</tr>
<tr>
<td>GR</td>
<td>.04</td>
<td>.04</td>
<td>.85</td>
<td>.85</td>
<td>.85</td>
<td>.85</td>
</tr>
</tbody>
</table>

Left: Table 4. Construct correlations & the square root of average variance extracted (AVE, bold).
Right: Table 5. Cronbach’s alpha (Cα), composite reliability (CR) & average variance extracted (AVE).
In the following, we check the formative indicators (i.e., response efficiency and response extensiveness). We followed Petter et al. (2007) for assessing the validity and reliability of the formative constructs. The traditional evaluation criteria such as factor loadings and AVE are not applicable for evaluating formative measurement models. Because these measures assume high internal consistency (high intercorrelating indicators), they are inappropriate for formative indicators, where no theoretical assumption is made about inter-item correlation (Petter et al. 2007; Straub et al. 2004). We assessed construct validity by using principal components analysis to examine the item weights for the measurement model (Petter et al. 2007). Table 7 depicts the inter-item and item-to-construct correlations for the formative indicators. If the majority of inter-item correlations and item-to-construct correlations for a given latent construct are significant, the measures achieve convergent validity. As can be seen, all items correlate with one another within the same construct higher than with items of the other construct with only a few exceptions which indicates discriminant validity. While only one item (RE3) shows a significant outer weight (see Table 9), and one might drop the other items, all items share more variance with their respective construct than any other item which is why we keep the items (Cenfetelli and Bassellier 2009). Furthermore, small absolute and insignificant weights should not inevitably be misinterpreted as a poor measurement model (Chin 1998; Hair et al. 2011). Following this analysis, we assume convergent validity.

In line with Lee and Xia (2010), RX3 showed an above-threshold (Diamantopoulos and Siguaw 2006) value for the variance inflator factor (VIF, see Table 8). As Lee and Xia (2010) argue, removing RX3 would not benefit the measurement and the item was retained. Therefore, we ensure that no multicollinearity issue exists and that reliability is given.

<table>
<thead>
<tr>
<th></th>
<th>RX1</th>
<th>RX2</th>
<th>RX3</th>
<th>RX4</th>
<th>RX5</th>
<th>RX6</th>
<th>RX</th>
<th>RE1</th>
<th>RE2</th>
<th>RE3</th>
<th>RE4</th>
<th>RE5</th>
<th>RE6</th>
<th>RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX1</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX2</td>
<td>.58*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX3</td>
<td>.66*</td>
<td>.70*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX4</td>
<td>.72*</td>
<td>.76*</td>
<td>.72*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX5</td>
<td>.50*</td>
<td>.60*</td>
<td>.51*</td>
<td>.43*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX6</td>
<td>.51*</td>
<td>.56*</td>
<td>.67*</td>
<td>.52*</td>
<td>.58*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX</td>
<td>.80*</td>
<td>.86*</td>
<td>.87*</td>
<td>.84*</td>
<td>.74*</td>
<td>.79*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE1</td>
<td>.40*</td>
<td>.24*</td>
<td>.43*</td>
<td>.275</td>
<td>.45*</td>
<td>.52*</td>
<td>.47*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE2</td>
<td>.29*</td>
<td>.49*</td>
<td>.51*</td>
<td>.44*</td>
<td>.13</td>
<td>.27</td>
<td>.44*</td>
<td>.39*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE3</td>
<td>.59*</td>
<td>.50*</td>
<td>.61*</td>
<td>.62*</td>
<td>.27</td>
<td>.45*</td>
<td>.59*</td>
<td>.40*</td>
<td>.60*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE4</td>
<td>.66*</td>
<td>.53*</td>
<td>.45*</td>
<td>.63*</td>
<td>.23</td>
<td>.32</td>
<td>.57*</td>
<td>.29</td>
<td>.70*</td>
<td>.60*</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE5</td>
<td>.25*</td>
<td>.36*</td>
<td>.40*</td>
<td>.39*</td>
<td>.41*</td>
<td>.41*</td>
<td>.45*</td>
<td>.54*</td>
<td>.46*</td>
<td>.62*</td>
<td>.26</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RE6</td>
<td>.33*</td>
<td>.43*</td>
<td>.31</td>
<td>.35*</td>
<td>.21</td>
<td>.26</td>
<td>.38*</td>
<td>.34</td>
<td>.62*</td>
<td>.39*</td>
<td>.55*</td>
<td>.43*</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>RE</td>
<td>.55*</td>
<td>.56*</td>
<td>.58*</td>
<td>.59*</td>
<td>.37*</td>
<td>.49*</td>
<td>.64*</td>
<td>.65*</td>
<td>.84*</td>
<td>.81*</td>
<td>.74*</td>
<td>.75*</td>
<td>.73*</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 7. Inter-Item and Item-to-Construct Correlation Matrix for Formative Items.

* p < .05; EX: Response Extensiveness, EF: Response Efficiency

Having established our model as valid and reliable, we can now turn to our hypotheses. As can be seen in Figure 2, all hypothesized paths but one are significant. Hypotheses H3 and H5 cannot be accepted. As diversity does not increase affective conflict in our sample but actually decreases it, we cannot accept H3. Regarding the moderating effect of temporal leadership on the effect of time pressure on affective conflict, we cannot accept the hypothesis, as it is not significant (p = .080).

Table 7 displays $R^2$ values, the achieved power as well as the effect sizes and respective $f^2$ values. As can be seen, most constructs provide medium or even large effects and high power in spite of the small sample size of 33.
DISCUSSION & CONCLUSION

We were able to support most of our hypotheses. Large effect sizes, high power, and high explained variance indicate that our small sample size of 33 participants is sufficient to draw meaningful conclusions. The model suggests that AISD is affected by requirements changes in regard to time pressure, affective conflict, and ultimately performance. Furthermore, the results suggest that diversity can reduce affective conflict and therefore mitigate some negative effects of time pressure, and that response efficiency, response extensiveness, as well as performance are linked to temporal leadership.

In contrast to our expectation, diversity decreased affective conflict in our sample. We assume this to be due to a medium level of diversity present in our sample. Age (SD = 2.18), experience (SD = .66), and educational background (SD = .55) were more homogeneous than in many AISD teams, as all participants were undergraduate students from the same university. This homogeneity might have reduced the likelihood of affective conflicts by providing a common ground and a feeling of belonging to the in-group (Brewer 1979). Another explanation would be that the in-group was perceived as more complex and diverse than other groups (Park and Rothbart 1982) and therefore, due to the self-reported characteristic, distort the measurement.

Another unexpected result was that no moderation of temporal leadership could be identified for the effect of time pressure on affective conflict. Another limitation and explanation for this might be that this sample was set in an educational context. While the class is designed to prepare students for real-life projects in cooperation with real-life practitioners, we cannot rule out a distortion. This might also have reduced the effect of affective conflict due to a sense of community among students (Cheng 2004; Cicognani et al. 2008).

Therefore, we call for replications in real-world scenarios with larger sample sizes. Replications would not only increase the reliability, but also increase the transferability to other contexts. Due to the small sample size and the participants being students, generalizability is not given. Future research could also investigate, whether or not the self-report items on diversity significantly deviate from the actual diversity to clear the mixed results regarding diversity. Furthermore, we measured different projects but only over one iteration. Including multiple iterations might result in further insights regarding the effect of time pressure.

Regarding theoretical contributions of this paper, we found supporting evidence for the importance of time pressure and temporal leadership for performance as outlined by previous research. Further, we integrated diversity and requirements changes to provide a more holistic view and are able to reduce the ambiguity of effects regarding diversity, requirements changes, and performance.

In regard to practical contributions, we laid the groundwork for a deeper understanding of the effects at play in AISD teams which are operating under constant requirements changes. The proposed model helps practitioners compose and steer teams to increase a team’s resilience towards requirements changes and time pressure.
In this paper, we presented the results of a quantitative study. We added explanation to the effects and interactions between performance, requirements changes, time pressure, affective conflict, diversity and temporal leadership. We provided reasoning for the hypothesized relations and argued for deviations and discussed the contributions to both research and practice and derived future research opportunities from the limitations of this study.

REFERENCES


Spanning Knowledge Holes In IS Projects

Gloria H.W. Liu  
National Central University  
glorialiu2007@gmail.com

Cecil E.H. Chua  
The University of Auckland  
aeh.chua@auckland.ac.nz

Eric T.G. Wang  
National Central University  
ewang@mgt.ncu.edu.tw

Abstract

Prior studies have demonstrated the importance of bridging structural holes across functional groups in IS projects. In this study, we argue that bridging structural holes is necessary but insufficient for ensuring project success. An additional requirement is that knowledge holes across functional groups need to be bridged to enable effective problem-solving across functional groups. We propose and empirically study the concept of knowledge holes in a case study of an ERP upgrade. Our findings suggest that complementary to the concept of structural holes, the concept of knowledge holes is useful for explaining different project outcomes. Our findings also demonstrate methods for bridging knowledge holes. Contributions of this study are manifold.

Keywords

Knowledge holes, structural holes, boundary spanner, IS projects.

Introduction

The proliferation of knowledge communities and the need for their integration result in the emergence of flexible organizational forms, such as cross-functional projects (e.g., ERP implementations). During cross-functional projects, people representing different communities coordinate to achieve organizational goals. However, there is no guarantee that by assigning people to projects, knowledge will be transmitted/created for achieving goals.

This paper introduces the idea of knowledge holes. The literature has documented the important role of boundary spanners who use weak ties to bridge structural holes across organizations’ functional boundaries (Hargadon and Sutton, 1997). The boundary spanner literature has generally focused on their relationship structures and how they help bridge structural holes. We show that boundary spanners must also bridge knowledge holes (Pawlowski and Robey, 2004). Each function contains within it specialized knowledge. When boundary spanners perform their role, some knowledge is transmitted across the boundary. We show such knowledge does not translate successfully unless boundary spanners understand the knowledge within the transmitting boundary. Thus, boundary spanners must span not only the structural hole, but the knowledge hole.

We demonstrate our claim through a cross-case analysis of two departments participating in an ERP upgrade project. In one case, boundary spanners between IT and the function understood knowledge across the functional boundary, including syntax/meanings/consequences. Thus, solutions were localized around problems faced in the department, and jointly implemented. In the other, this knowledge was poorly understood resulting in incomplete solutions and unresolved problems. Our contribution is a more nuanced theorization for analyzing events and actions within and across IS project boundaries. Our findings further suggest a shift of focus from boundary spanners’ appropriating boundary objects for bridging knowledge holes (Carli, 2002, 2004) to their practices and interaction for creating common narratives about those events and actions.

Knowledge spaces of an IT project

Cross-functional IT projects involve designing/implementing IT artifacts where at least two organizational departments are involved (Simon and Newell, 1971). In most such projects, one cross-boundary problem occurs where the project team (i.e., internal IT members and/or consultants) must understand the non-IT problem (i.e., the problem space) and construct IT solutions to the problem (i.e., the design space) (Purao, Rossi and Bush, 2002).

The problem space is a metaphorical space containing the team’s interpretation of user requirements in the face of a task environment (Purao et al., 2002). It includes a mental model of “a subset of the real world with which a computer
system is concerned” (Guindon, 1990: 317). To construct the problem space, there must be an understanding of the task/requirements, and a conceptual match between information about the task/requirements and IT technologies’ capacities/advantages/limitations/impact (Bassellier, Reich and Benbasat, 2001). Conceptual mismatch can cause major difficulty in constructing the problem space.

The design space, also known as the implementation domain (Blum, 1989), is a metaphorical space that contains mental representations of the team’s solutions to the problem, based on which the team creates formal models/specifications for building systems (Purao et al., 2002). The team can explore diverse solutions based on current IS methods/techniques (Oxman, 1997). For example, information systems can be developed in-house, outsourced, or customized from application packages. Or an IT project can follow open source or agile development. Different solutions can generate distinct consequences. The project team and departments involved share the consequences of a solution (Carlile, 2002, 2004).

The knowledge domains in the problem and design space are often different (Iivari, Hirschheim and Klein, 2004). The knowledge domain of the problem space (i.e., application domain knowledge) is often associated with the departments/functions. For example, in an accounting-based IT project, lots of the necessary problem-space knowledge will be associated with accounting. The literature has emphasized the importance of application domain knowledge for solving problems in the real world. High application domain knowledge is found to prompt IT teams to engage in strategies contingent upon the nature of a problem: a focused search for solving simple problems, and an exploratory search for complex, ill-defined problems; in contrast, teams with low application domain knowledge tend to be distracted by simple problems’ surface features (e.g., the order of prompts in a problem description), and cannot meaningfully code information for solving complex problems (Khatri and Vessey, 2016).

In contrast, the knowledge domain of the design space is often technical/IT related (i.e., IS domain knowledge). Algorithms, hardware configurations, programming languages, design languages and databases are often associated with the design space. Technical complexity can be a barrier for departments to understand IT’s language/meanings, and to envision consequences associated with solutions. The literature has documented the way IT projects and their impacts is communicated to departments is consequential (Lapointe and Rivard, 2005). Further, departments’ knowledge for designing workarounds and understanding of costs/benefits/risks associated with operating an IS can impact their choices about how to engage with the system, leading to organization-wide consequences (e.g., errors/inefficiencies/shadow systems) (Alter, 2014). Given the two kinds of knowledge are possessed by different departments, it is often necessary to have individuals perform boundary spanning to bridge departments.

**Boundary Spanning**

Boundary spanning refers to activities, occurring at functional boundaries (Pawlowski and Robey, 2004). It is about creating/maintaining linkages to “monitor, exchange with, or represent” (Mange and Eisenberg, 1987: 313) a group to its environment. Boundary-spanning can be responses to the environment, or proactive moves for managing interdependencies (Cross, Yan and Louis, 2000).

It involves a two-step process: searching out relevant information on one side, and disseminating it on the other (Tushman and Scanlan, 1981). Each department only knows things under its purview, so departments need information from others to adapt/coordinate goals and activities to meet organizational/environmental demands. However, searching without disseminating creates internal silos (Roberts and O’Reilly, 1979). Thus, successful boundary spanning must also fulfill the external representation function to important outsiders, such as customers/suppliers/the board of directors for obtaining their support/resources (Ancona and Caldwell, 1988). Boundary spanning can only be accomplished by those who are well connected externally/interally. Specific boundary spanning activities include environmental scanning, contractual negotiation, task coordination (Choi, 2002), building relationships (Druskat and Wheeler, 2003), representing projects to stakeholders (Marrone, 2010), and routinizing information searching/acquiring/storing activities (Hargadon and Sutton, 1997).

The boundary spanning literature principally employs network structure as a proxy for information flow and assumes connections lead to information processing across boundaries (Granovetter, 1973, 1983; Hargadon and Sutton, 1997; Podolny and Baron, 1997; Xiao and Tsui, 2007). How the substance of the network structure is moved/combined/transformed across structural holes is rarely studied. An exception is the work of Pawlowski and Robey (2004) who argue that boundary spanners need to reframe/translate information from one group in terms of the
perspective of another, deliberately ask why to challenge current processes, and build cases to generate support for their proposals.

**Knowledge Hole**

Structural hole theory highlights the information and control benefits boundary spanners can create in an ill-connected network (Burt, 1992, 2005). Boundary spanners are often the main channel to access knowledge and to negotiate solutions across boundaries. Their network position exposes them to information that reveals conditions/opportunities otherwise invisible to those within boundaries. That information can possibly inform strategies to negotiate solutions. A structural view of boundary spanning thus must be augmented by considering the knowledge understood by boundary spanners in the network. Other research finds while knowledge diversity is correlated with network structure, there is considerable variance unexplainable by network structure (Rodan and Galunic, 2004). We therefore argue just having individuals perform boundary spanning is insufficient to bridge the problem space/design space gap. We introduce the concept of knowledge holes, arguing that knowledge holes should be spanned so that knowledge can be applied on both sides of the hole to solve shared problems. Knowledge holes refer to the absence of shared syntax/interpretations/consequences that impedes problem-solving across boundaries. They can be considered as a complement to “structural holes” which are missing relations that inhibit information flow between people (Burt, 1992).

Based on Carlile (2002; 2004; Carlile, 2002), we argue knowledge across boundaries comprises three elements: shared syntax/interpretations/consequences. First, a shared syntax is a medium for representing/storing/retrieving knowledge with fixed meaning (Boland and Tenkasi, 1995; Vilhena et al., 2014). It is vocabulary specific to situations (Khatri, Vessey, Ramesh, Clay and Park, 2006). For example, the ER model’s symbols (e.g., rectangles/diamonds) are the syntax to represent objects, abstract concepts and their relationships in a system. While shared syntax is always necessary to analyze problems, it is insufficient to represent semantic differences and dependencies, particularly when novel conditions emerge (Boland and Tenkasi, 1995; Carlile, 2002). Even under stable conditions, the same concept can have different connotations for different people. For example, the concept of production cost means different things for the accounting and manufacturing departments. For accounting, production cost involves calculating the accurate actual cost with consideration of equipment/machinery depreciation. For manufacturing, production cost emphasizes the variance analysis between predefined and actual cost for monitoring/intervention.

Second, shared interpretation means there is consensus of meaning (Boland and Tenkasi, 1995; Carlile, 2002). Shared interpretation is situated in the context and contains a group’s systems of meaning and cognitive repertoires, i.e., what they know and how they know it. It cannot be easily transferred across boundaries, and requires translation into another group’s perspective (Carlile, 2004). Shared interpretation can only be reached by understanding the nuances/details of actual practice (Brown and Duguid, 1991), learning from different interpretive communities (Fish, 1980), or enhancing mental models by using cognitive support tools (Vitharana, Zahedi and Jain, 2016). Shared interpretation recognizes even if shared syntax exists, interpretations can be different and evolve over time/space (Carlile, 2002). For example, through interacting with users, a team discovers “data availability” not only means data available “at users’ request” but “the liberty” to retrieve/update data when needed.

Third, shared consequences recognizes the purposive nature of knowledge as people create/apply knowledge to solve problems (Carlile, 2002). Shared consequences involve developing common interests and making trade-offs between actors (Brown and Duguid, 1991). Common interests motivate joint problem-solving, whereas when interests are in conflict (i.e., solving a problem does good for one, but does harm for the other), one of the parties may be unwilling to make changes; likewise, projected positive consequences of a solution motivate people to adopt the solution, whereas negative ones imply the need to alter the solution or create a new one, and validate it (Carlile, 2002, 2004). Thus, shared consequences can be achieved by identifying actors involved, convincing them they have common problems, and persuading them to accept responsibilities and outcomes associated with the solution (e.g., learning or transforming skills/knowledge) (Callon, 1986).

Constructing the problem space/design space requires shared syntax/interpretations/consequences, so that common issues and potential solutions can be identified/debated/understood. Problems not represented, translated and resolved can prove consequential over time. For example, many ERP workarounds are performed because the problem being worked around is not represented in the system. Likewise, solutions not understood, negotiated and valued will not receive attention and support needed to implement them. For example, users who are not well trained may ignore an ERP system’s querying facilities in favor of doing their own analysis in MS Excel.
As we move from shared syntax, to shared interpretations, and to shared consequences, complexity increases. Shared consequences require the existence of shared syntaxes and interpretations (Carlile, 2004). However, it is also likely that given shared syntaxes and interpretations, actors are unwilling to make trade-offs and negotiate solutions/responsibilities, because transforming or learning new skills can be costly.

Table 1 presents our preliminary conception of the intersection between the problem/design space and knowledge holes (Liu, Chua and Wang, 2016). A hypothetical knowledge failure in any of the quadrants could lead to difficulties implementing IT projects.

<table>
<thead>
<tr>
<th>Problem Space</th>
<th>Design Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactical hole</td>
<td>Definition: IT’s failure to comprehend terms/labels that describe the problem a department faces in their task environment. Example: IT is not clear about what the consolidated financial statement is composed of, such as which companies are subsidiaries or associates, which accounting items are included, which currencies are used etc.</td>
</tr>
<tr>
<td>Definition: departments’ failure to comprehend terms/labels that describe IT solutions. Example: Accounting thinks of data storage as a group of Excel spreadsheets and does not appreciate the additional complexity of querying a database. They thus neglect to specify important data cubes they need</td>
<td></td>
</tr>
<tr>
<td>Interpretation hole</td>
<td>Definition: IT’s failure to adjust their interpretation of the problem a department faces in related contexts. Example: IT fails to understand that “price” is negotiated between buyer and seller and assumes everything has a fixed price under all conditions.</td>
</tr>
<tr>
<td>Definition: departments’ failure to evaluate potential IT solutions and their implications. Example: Purchasing is not aware that input is required from them for IT to develop solutions to effectively integrate with suppliers.</td>
<td></td>
</tr>
<tr>
<td>Consequence hole</td>
<td>Definition: IT’s failure to envision how a problem influences the departments involved and agree on the scope of the problem. Example: IT understands a requirement, but thinks of the requirement as of low priority to be delayed to the next implementation cycle. They don’t understand not implementing this violates accounting principles.</td>
</tr>
<tr>
<td>Definition: departments’ failure to envision how the adopted IT solution impacts the departments and accept ensuing responsibilities and outcomes. Example: Marketing thinks of an IT implementation as a new physical device used by data-entry people. They don’t understand the new system will impact those who aren’t using the system (e.g., by impacting commission calculations).</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Syntactical/interpretation/consequence holes in the problem/design space

Methodology

We conducted a cross-case analysis of a 10-month long ERP upgrade project in two functions of a Taiwanese manufacturer (ElectroCom) (Yin, 2003). An upgrade is the replacement of an installed version with a new one from the software vendor (Khoo and Robey, 2007). It varies in terms of scope (technical and/or functional upgrade) and version (minor or major) (Ng, 2001). The project involved a major version and functional upgrade of a highly customized system. Hence, bringing together diverse knowledge was needed to decide what the upgraded system would look like. We compared how knowledge domains (i.e., application/IS domains) understood by representative users (i.e., boundary spanners) in the two functions affected the construction of the problem/design space, and ultimately the IT project outcome for the departments concerned.

Research site

ElectroCom is a Taiwanese manufacturer, with headquarters in Taiwan, factories in Taiwan and China, and sales offices across Pacific Asia/Europe/the US. At the time of study, it employed over 2,500 employees worldwide.

An ERP system had been used across ElectroCom in Taiwan and China since its first implementation in 1999. The upgrade project was expected to affect offices and factories in Taiwan and China. This project was in response to the local Taiwanese government’s adoption of International Financial Reporting Standards (IFRS) two years later (i.e., in 2013). It upgraded the database (from Oracle 9i to 11g) and ERP version (R11.5.10 to R12.1.3) from Oracle EBS.
The upgrade was scheduled for 10 months. However, because of problems encountered, the upgrade for the marketing division was delayed by 3 months. Three modules (finance, sales and distribution, and production) were upgraded across 4 divisions (SCM, marketing, general administration, and manufacturing).

The upgrade project was the largest IT project in ElectroCom at the time, involving 6 internal IT members and 4 consultants. The internal IT members were highly skilled (an average company tenure about 10 years), with the most junior one having work experience with the Oracle ERP system for about 5 years. The consultants were hired principally to facilitate training. The ERP upgrade cost approximately 150 thousand US dollars, including an annual fee for the maintenance/support from Oracle, and the training fee paid to Taiwanese consultants. The amount excludes new hardware, internal IT personnel costs, and the opportunity costs of users participating in the project.

Data collection

The first researcher accessed the site about 1 year after project completion to collect retrospective data. We collected data from multiple sources (management/non-management Consultants) and used multiple methods. Data collection methods include (1) interviews, (2) documentation, and (3) on-site observation (Table 2). The documentation, especially, helped combat the retrospective nature of data collection as document contents do not change over time.

<table>
<thead>
<tr>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project proposal</td>
</tr>
<tr>
<td>Minutes of meetings including the kickoff and review meetings</td>
</tr>
<tr>
<td>Project schedule</td>
</tr>
<tr>
<td>Project-related training materials</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On-site observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviews</td>
</tr>
<tr>
<td>Stakeholders</td>
</tr>
<tr>
<td># of interviews</td>
</tr>
<tr>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Top management (including CIO)</td>
</tr>
<tr>
<td>SCM representatives</td>
</tr>
<tr>
<td>Marketing representatives</td>
</tr>
<tr>
<td>IT</td>
</tr>
<tr>
<td>Consultants (project manager)</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Table 2. Breakdown of data sources

We first queried two knowledgeable IT members (IT project manager-CIO and project coordinator) about divisions affected by the implementation. Two departments (marketing/supply chain management) had the strongest differences in outcomes. Specifically, the implementation in the marketing department was described as “a total disaster” and “appalling” (IT project manager), whereas the supply chain management (SCM) department described the new ERP as “richer” and the project helped “connect more dots” (SCM user). The first author thus focused data collection on those two departments to observe contrasting implementation processes. As data collection proceeded, we serendipitously discovered representative users’ (i.e., boundary spanners) understanding of the IT domain knowledge affected the construction/implementation of IT solutions (i.e., bridged vs. unbridged knowledge holes).

During data collection, the first author was assigned a meeting room in ElectroCom’s Taiwan premises to conduct interviews. Interviews with two former employees and one consultant who were key project participants were conducted outside of the company premises.

We developed an interview protocol and adapted it to reflect interviewees’ positions and issues as the research progressed. Interview questions focused on issues related to project management (e.g., planning/execution/control and coordination/problem-solving/evaluation). We asked interviewees (1) their roles in the organization/project, (2) tasks they involved, and (3) their experiences/perceptions in the project.

Data analysis

Within each case, we asked initial interviewees to identify potential boundary spanners who had more interaction across departments. Representative users (i.e., key users) from both functions were nominated as boundary spanners.
They were required to acquire knowledge about user needs, relay knowledge to IT, and help implement IT solutions. Based on preliminary definitions of knowledge holes in Table 1, we then focused on coding the three types of knowledge holes in the problem space/design space. Table 3 has sample quotes. New concepts were also allowed to emerge, and were categorized. These new codes captured contextual factors associated with knowledge holes, causal mechanisms explaining bridged/unbridged holes, and project outcomes. The analysis followed the constant comparison logic (Eisenhardt, 1989; Yin, 2003).

<table>
<thead>
<tr>
<th>Code definition &amp; Quote</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntactical hole</strong></td>
</tr>
<tr>
<td>(problem space) Syntactical holes existed (1) if there were no common terms/labels to describe problems/concerns departments faced, or (2) if IT failed to understand terms/labels departments used to describe their problems/concerns; syntactical holes were bridged if IT shared and understood terms/labels that departments used to describe problems they faced.</td>
</tr>
<tr>
<td>Many obsolete data [suppliers who they stopped trading with] were still around... They were becoming a burden to the system. (Consultant) [hole bridged because both IT and department described data of suppliers who they stopped trading with as the major problem]</td>
</tr>
<tr>
<td>(design space) Syntactical holes existed (1) if there were no common terms/labels to describe IT solutions, or (2) if departments failed to understand terms/labels IT used to describe solutions; syntactical holes were bridged if departments shared and understood terms/labels IT used to describe IT solutions.</td>
</tr>
<tr>
<td>IT told us the changed data structure a bit...we didn’t discuss much about the structure...[we’re] thinking that this would be just another minor upgrade... (Marketing key user) [holes not bridged because the department did not grasp the connotation of a changed data structure]</td>
</tr>
<tr>
<td><strong>Interpretation hole</strong></td>
</tr>
<tr>
<td>(problem space) Interpretation holes existed, if IT failed to adjust their interpretation of the problem a department faced in different settings; interpretation holes were bridged when IT had a mental extension or an awareness of contingencies that could change the interpretation of the problem.</td>
</tr>
<tr>
<td>…each of [the users] dealt with distinct suppliers. Some were in charge of purchasing raw materials, and some in charge of purchasing tools or equipment...They were exclusive contacts for distinct suppliers. (IT, manufacturing module) [hole bridged because IT understood contingencies, i.e., data ownership, that might require alteration of IT solutions]</td>
</tr>
<tr>
<td>(design space) Interpretation holes existed, if departments could not evaluate potential IT solutions and their implications according to their local situations; interpretation holes were bridged when departments had a mental extension or an awareness of contingencies that could possible change the meaning of potential IT solutions.</td>
</tr>
<tr>
<td>Users complained about why IT couldn’t just migrate the data and save them efforts and time. I explained even if IT could do so, the changed data structure meant we had to check the data [manually]...(SCM key user) [hole bridged because the department could evaluate alternative IT solutions according to local situations and their implications]</td>
</tr>
<tr>
<td><strong>Consequence hole</strong></td>
</tr>
<tr>
<td>(problem space) Consequence holes existed, if IT failed (1) to understand how a problem or project task influenced the department, or (2) to reach an agreement on the problem/task scope; consequence holes were bridged, when IT (1) understood the impact of a problem/project task to departments, and (2) reached an agreement on the problem/task scope.</td>
</tr>
<tr>
<td>…We helped migrate data of clients and orders...[users] filled extra fields manually...it’s just garbage in, garbage out. (IT, distribution module) [hole not bridged because IT was only willing to map out part of the problem scope and offered a partial solution]</td>
</tr>
<tr>
<td>(design space) Consequence holes existed, if departments failed (1) to understand how adopted solutions impacted the departments and (2) to accept ensuing responsibilities and outcomes; consequence holes were bridged, when departments (1) understood the impact of adopted IT solutions, and (2) accepted ensuing responsibilities/outcomes.</td>
</tr>
</tbody>
</table>
...To enter that amount of data... I kind of underestimated efforts required... also some sales orders were to be re-entered and checked... torrential workload in the last week... (Marketing key user) [hole not bridged because the department underestimated their responsibilities associated with adopted IT solutions]

Table 3. Representative quotes grouped according to codes

Due to space constraints, we present only one problem that provided us with rich insights (i.e., data management problem). The problem in the SCM division is counterbalanced by a similar problem in the marketing division. This contrast demonstrates how knowledge holes were (not) bridged. For each case, we highlight the knowledge holes in the problem/design space.

Findings

When the project began, key users from all departments attended training sessions about the new ERP. They were introduced to the ERP’s new features and the new financial regulations. Initially, they thought the project was mainly to “help the finance people out” (Marketing key user) and knew “some effortful participation” (SCM key user) was required from them.

The SCM and marketing key user had worked with IT on other projects before. They were trusted by the department head and empowered to make decisions associated with the project.

Master data quality

Master data is a single source of common data shared across systems/applications/processes. It describes attributes of business entities (e.g., products/sites/clients/suppliers), and is rarely changed. Different business domains thus can use master data for their own functional needs. Managing master data to enable the use of accurate, timely, and relevant data across systems/applications is essential (Spruit and Pietzka, 2015).

ElectroCom had accumulated a large amount of master data in the ERP over time, and protected it from unauthorized changes. The common narrative was the master data constituted “the foundation” (SCM key user) and “a critical point of control” (SCM user) for many IT applications/systems. However, the new ERP’s data structure was more complex. It had more fields to fill, and was presented differently (HTML pages vs. a Windows-based form).

Vignette: supplier master data

Bridging knowledge holes in the problem space: After the kick-off, IT informed the SCM key user about the task of managing the master data (e.g., cleansing/updating/enriching). The key user then expressed her concern about users’ unwillingness to do the task, especially if users could not foresee its benefits. After the discussion, the key user and IT agreed the major problem with the supplier master data lay in its obsolescence which could cause inefficiencies in daily routines and slow down the system. “Obsolescence” became the buzzword used by the SCM and IT divisions to describe the problem.

Many obsolete data [suppliers who they stopped trading with] were still around... They were becoming a burden to the system. (Consultant)

The key user further probed to identify areas in supply chain the new data could possibly improve. She discussed with IT/consultants during training and informally. IT thus understood the key user anticipated the new data would be applied to enhance supply chain’s analytical capability.

... [the SCM key user] asked consultants lots of questions... how those new data could possibly be useful... how to use those data to streamline their process... improve their analysis... we communicated a lot informally via phone or in person... (Project coordinator)

Because of the conversations, the team realized the data-entry interface might compromise data quality, which would then reduce data analysis quality. The key user identified specific problems, including the lack of user familiarity with the interface and the language barrier (i.e., Taiwanese users using an English interface). IT thus could realistically
imagine the specific difficulties users would face, and understood that users “needed some process to ensure its quality” (IT, manufacturing module).

The key user also elaborated how users coordinated daily tasks in the SCM division. She made explicit her concern that specific users would be overwhelmed by additional data-entry tasks. Also, she highlighted unless something was done, these data-entry users would be unable to obtain other SCM users’ support. The key user thus lobbied IT to allow users to directly modify certain master data elements rather than seeking changes through the organizational bureaucracy.

The key user explained to me that users deal with distinct suppliers…They are exclusive contacts for distinct suppliers. (IT, manufacturing module)

**Bridging knowledge holes in the design space:** IT and users shared assumptions about the master data, including that it was “the foundation” (SCM key user) and “a critical point of control” (SCM user). IT initially confined the master data task to the key user. However, the key user formed a team to sell the project across the division and roll out training for managing master data to users.

...explained to [3 relatively junior users] what I saw in this project...we four acted like a gang, demonstrating what to do to other users... (SCM key user)

Only [the key user] and I attended the training session [for managing master data]. We then came up with our individual versions of SOP [standard operating procedures]. Each of us then taught another user one-on-one, asking the user to follow the SOP to input data of two suppliers...and wrote up their own SOPs...after this, these two users showed their SOPs to other two users and taught them...the teaching and learning snowballed from two to four, to six, to all users... (SCM user)

Due to the training/guidance users received, they understood the data structure and access methods. IT agreed to extend SCM users’ access to the master data. Seven SCM users participated. IT proposed two routes for solving the problem: (1) IT would migrate data to the new ERP, and users would update/correct data; or (2) users would compile data in Excel spreadsheets and then copy/paste the data to the new system. The key user realized the first option would create more risk. The key user and IT used the metaphor of house renovation vs. house building to explain the options to users.

...I explained even if IT could do so [migrated data on users’ behalf], the changed data structure meant we had to check the data [manually]...it’s like building a new house vs. renovating one in very bad condition...we agreed with the “house building” solution. (SCM key user)

The key user was mindful the increased workload could be a point of resistance with users. She identified users with more critical jobs, and decreased their responsibilities.

To achieve desired data quality, the key user helped develop procedures (e.g., feedback loops) and solicited management support to remove distractions.

...spending one whole week...in a meeting room without disruption...users formed 2-person groups...both entered data of their suppliers, and had the other check the accuracy... (SCM key user)

Consequently, SCM users accomplished the master data task on time and with high quality, and solved a “recurrent problem with converting PR [purchase requests] into PO [purchase orders]” (SCM user).

**Counter-vignette: Client master data**

**Knowledge holes in the problem space:** In the beginning of the project, IT informed the marketing key user about managing client master data. IT warned the key user about the changed data structure and attendant risks (e.g., mismatched data). Because the marketing key user perceived the system would provide minimal enhanced functionality to marketing, she considered the upgrade as “another minor upgrade” and the master data issue purely as a data-entry task for solving no specific problems. She did not further probe why new data would be needed, how it could be useful, and how historical data could be better managed.
Users didn’t see many improved system functionalities...didn’t fully grasp associated changes underneath... (IT, distribution module)

Instead, because of the tremendous amount of client data, the marketing key user was more concerned if they could finish this task within an assigned deadline and in a consistent format. The issue of data quality was considered a luxury hard to achieve. In retrospect, the IT project manager described this as “a huge cognitive gap” between IT and marketing.

...new fields to be filled...in a consistent format...three persons would render three different formats...very limited time to verify data... (Marketing key user)

The marketing key user was the only person from the division to make decisions. To contain the project’s impact, she hoarded project information and assigned a junior administrative assistant to manage the data-entry task on her behalf.

The marketing key user worked in her silo. She did not consult those affected by the data-entry task. She did not communicate expectations/concerns of the new system with others. IT thus did not know activities users were involved in and underestimated the impact of the data-entry task on their routines.

...We told them about this early, hoping they could coordinate their efforts...we didn’t find the efforts were seriously underestimated until very late...kind of too late to react at the end... (IT, distribution module)

**Knowledge holes in the design space**: IT saw the project as “a reimplementation of a new ERP” (IT project coordinator) in which master data should be reviewed/maintained. However, the marketing key user expected the upgrade to maintain status quo, and was satisfied “as long as data didn’t cause processes to stop.” Therefore, training was not taken seriously, and knowledge not mapped to daily practices.

**IT would tell us to note down something...some terms I didn’t even understand at that time...** (Marketing key user)

In addition, the key user considered users as a threat to data consistency, and wanted to minimize their participation. IT thus suggested automatic migration of client master data to the new ERP, and one designated user to manage the data. The key user sent the junior administrative assistant to enter and verify the data. However, the assistant received limited training prior to starting the task. She did not understand the importance of the master data, and had little knowledge about on-the-ground processes.

...I mainly learnt the data structure on-the-job...Later, another admin assistant was assigned to join the task...she didn’t know much about information systems. (Marketing administrative assistant)

The marketing key user adopted the IT solution as is without considering its impact on the assistant and the support the assistant needed. Due to the lack of other users’ support/cooperation, the assistant could only perform the task perfunctorily. IT described this situation as “garbage in, garbage out” (IT, distribution module)

...we sent checklists to sales assistants to ask for their help to verify data...not many of them replied... (Marketing administrative assistant)

Thus, the task was delayed and quality was below par. The junior assistant and several sales assistants were forced to work during national holidays and experienced tremendous stress.

Table 4 summarizes knowledge holes and boundary spanning activities associated with the data management problem.
Our study of an ERP upgrade project across two divisions in a large manufacturer reveals the importance of bridging knowledge holes for cross-functional IT projects. Within the ERP upgrade project, key users and IT (i.e., IT representatives for individual modules) brought in diverse knowledge, and served the role as boundary spanners. Research on structural holes shows boundary spanners are exposed to alternative ways of thinking, and are important for combining/synthesizing information across boundaries (Burt, 2005). The SCM division case particularly reveals how boundary spanners (the key user & IT representative) jointly synthesized information to develop common narratives in the problem/design space to guide or regulate their actions. The narratives were built on common syntax with intricacies of actual work practices (e.g., exclusive data ownership) and considerations of mutual interests (concerns of data quality/user workload). In contrast, the marketing key user and IT representative failed to integrate/interpret the information. They were overwhelmed by information and could not filter irrelevant information and identify important information for actions. Therefore, either wrong/incomplete problems were identified or wrong/incomplete solutions were imposed.

Our findings suggest bridging knowledge holes in the problem/design space is required for positive project outcomes. First, syntactical holes need to be bridged by common words/language in the problem/design space. The SCM key user and IT both recognized the problem was "obsolete data." Because the two were able to exchange information successfully, they could grasp the implications of this from the SCM users and develop plans to manage contingencies.
Specifically, they realized to make this work would mean opening up data so SCM users could self-query, and that SCM users would need to be coached to understand how updated master data would benefit them.

Being able to visualize these consequences, in turn required both the SCM key user and IT appreciated how the changes would affect the SCM department. This in turn meant they had to understand the words the other used, translate words when needed to communicate consequences, and use the right words to persuade SCM users to take attendant responsibilities (e.g., acquiring skills for managing data).

In contrast, when the same scenario played in the marketing department, IT did not appreciate how the data-entry task would affect marketing users. Similarly, the marketing key user did not appreciate the changed data structure’s implications. Therefore, simple solutions with damaging consequences (e.g., assigning untrained assistants to enter/verify data) were implemented. The failure of both IT and marketing to appreciate the on-the-ground situation and consequences of their actions similarly stemmed from different language and understanding of each other.

Our findings thus confirm the importance of spanning syntax, interpretation, and consequence holes. Our findings also suggest ways to bridge said holes.

Our findings suggest one way of bridging the syntax hole is via using common words/labels to build common narratives. Common narratives connect characters with a sequence of events that have shared meanings (Cunliffe and Coupland, 2012). Within organization studies, narratives are found to be a means of making sense of (Boje, 1995) or giving sense to (Currie and Brown, 2003) a situation. In the SCM department, the key user and IT used common words (obsolete data, a critical point of control, house building) to construct narratives to describe the problem/design space. They thus could make sense of and give sense to the difficulties users faced (e.g., concerns of other users’ commitment) and IT solutions.

Second, at the interpretation level, more particularistic/local clues need to be integrated to the common narratives about the problem/design space. Through probing contingency factors and processes (discussed/walked through users’ data-entry procedures), the SCM key user and IT gained insight into users’ task environment, including direct, indirect, distal and near causes (e.g., system interface, task coordination, distractions) of some failure situations (e.g., data quality concern, perceived stress). IT thus could refine IT solutions accordingly (e.g., extensive data access for users).

Finally, consequence holes can be bridged by creating venues for stakeholders to negotiate problem scopes and learn new knowledge/skills. Because the SCM key user and IT routinely shared information, sought opinions from each other, and helped each other, a community of practice was formed beyond functional boundaries. The SCM key user also articulated her vision to users, arranged venues for their learning/working together, and consulted their opinions about decisions that would affect them. IT and SCM department thus could see mutual benefits in the project, collectively learn skills and knowledge, and value their new competence.

Our contributions build on a conceptualization of knowledge holes that is empirically studied in a real-life IS project. We demonstrate knowledge holes cannot be bridged before the syntactical, interpretation, and consequence holes in the problem/design space are bridged. Based upon Carlile (2002, 2004), our conceptualization provides a more nuanced theorization for analyzing events and actions within and across IS project boundaries. Our findings further suggest a shift of focus to boundary spanners’ active practices and interaction for creating common narratives about those events and actions.

Our findings concur with the foundational role of shared syntax in bridging knowledge holes (Carlile, 2004). Shared syntax can not only be used to transfer information accurately, but help develop common narratives with local details and considerations of mutual interests to regulate one’s thought/action.

Practically, this study explains how boundary spanners in big IS projects (e.g., key users/IT) may act to bridge knowledge holes. Our findings demonstrate boundary spanners bridge knowledge holes via building common narratives in the problem/design space. That is, boundary spanners use common labels/words/language to construct common narratives; integrate local clues from users’ work context to said narratives; and create venues for negotiating problems and learning new knowledge.
Conclusion

The proliferation of specialized knowledge communities highlights the importance of boundary spanning. In this study, we propose and empirically study the concept of knowledge holes in a case study of an ERP upgrade. Our findings suggest bridging knowledge holes is a necessary condition for positive project outcomes. The concept of knowledge holes thus complements the concept of structural holes for explaining distinct project outcomes. Conditions for bridging the knowledge holes is to bridge syntactical, interpretation and consequence holes separating functional groups in large IS projects.

References


Fish, S. E. (1980). Is There a Text in This Class?: The Authority of Interpretive Communities. Harvard University Press, Cambridge, MA.


Implementing Strategy Through IS Projects: A Theory Building Literature Review

Fred Niederman  
St. Louis University  
fred.niederman@slu.edu

Katherine M. Chudoba  
Utah State University  
kathy.chudoba@usu.edu

ABSTRACT

In project management, a growing area of importance is “benefits realization”, including techniques to achieve on time and on budget efficiency. Broadly, it refers to aligning efforts with strategic purpose, realizing the purposes for which the project was selected, and gaining benefits from its effectiveness. There is a dearth of writing about how strategy is realized through IS projects, particularly at the program or project level. Our purpose is to create a clear, more detailed and predictive linkage between organizational strategies and IS project enactments by addressing the question, “How do organizations translate strategy through IT-enabled strategic initiatives?” In this RIP paper, we lay the foundation for our review and describe the process for our examination. By examining mechanisms for executing strategy, findings should be relevant both to academics in terms of providing insights for further testing and/or refinement, as well as practice for forming a basis for predicting outcomes and selection of execution process techniques.

Keywords

Project management, IS project development, strategy, IS project outcomes

INTRODUCTION

Project management, information systems (IS), and strategy are domains that cross organizational functions. Researchers often draw on one domain to inform research in a second. For example, research at the intersection of project management and IS has found that even when project management best practices such as user engagement are followed, IS development projects are not always successful (e.g., Gallivan and Keil, 2003). Similarly, there is a large corpus around IS alignment with corporate strategy (e.g., Chakravarty, Grewal, and Sambamurthy, 2013; Setia, Sambamurthy, and Closs, 2008), and researchers have examined strategic management processes to formulate and implement strategy at the corporate level (i.e., Hill & Jones, 2001; Mintzberg and Quinn, 1996; Thompson, 2001). What is missing is how corporate strategy gets translated into implementation, particularly at the program or project level. In practice, the two sets of activities are well connected; projects and programs are important ways for strategy to be implemented in the enterprise and we ought to understand much better how this occurs (Kaiser et al., 2015). As expressed by Näsholm and Blomquist (2015, p. 60), “Programs are often of a strategic nature for the organization and have broad long-term implications (Artto and Kujala, 2008) and they can even be considered as tools for strategy implementation to achieve organizational objectives (van Buuren, Buijjs, and Teisman, 2010)”. Examining the intersection of these three domains offers the opportunity for new insights into best practices for more impactful IT-enabled projects and organizational strategy. The overarching objective is to address the question, “How do organizations translate strategy through IT-enabled strategic initiatives?”

In this research in progress (RIP) paper, we lay the foundation for our review and describe the process for our examination. In the background section, we elaborate on the goal of our investigation, define key terms, establish its boundaries, and present a set of guiding questions that expand on our research objective. Next, we briefly describe our methods, and then present the results of an initial and very preliminary analysis. We conclude with a description of the expected contribution of our review and theory building effort.

BACKGROUND

After an initial examination, we find no literature in any of the three key domains that directly addresses the question presented above. Ashurst and colleagues (2008) proposed the benefits realization capability model, which posits that...
benefits from systems implementation are derived from following a series of best practices, based on their evaluation of successful IT project implementations. Our interest is broader, including the initial decision to address organizational strategy through IT systems, and considering both successful and failed projects. The academic and popular presses are full of articles that describe failures across project management and strategy domains (e.g., Sull and Sull, 2015; Workfront, 2014). This is an enduring problem in practice, and an investigation across the three domains may yield theoretical and practical insights to address this problem (Lyttinen and Grover, 2017). The idea that projects are a mechanism for implementing organizational strategy is so deeply embedded in common thinking that it is rarely questioned or studied. As phrased by Pinto (2015) in a popular project management textbook: “Projects are building blocks in the design and execution of organizational strategies” (p. 6).

Teams working on particular tasks (e.g. doing projects) is intuitively a key to achieving strategic goals, but this is not necessarily the same as guiding projects to on-time and on-budget outcomes. Nor does it guarantee that projects will fulfill their ultimate objective to advance organizational strategy even when efficiently delivered (Hjelmbrekke, Laedre, and Lohne, 2014). Our purpose is to create a clear, more detailed and predictive linkage between organizational strategies and IT-enabled project enactments. We are particularly focused on the “how” question regarding how strategies can be effectively moved forward by IS projects. Assuming that no extant theories will be found that provide theoretical explanation and predictive power relative to the question of interest, we will draw on insights from our review and analysis of the literature to offer a theoretical model and set of propositions to guide future research.

This study uses two perspectives to examine implementation of strategy through IT-enabled projects. First, it examines the broader issue of what has been learned so far, particularly addressing whether there are mechanisms designed to implement strategy through projects and evidence about their effectiveness. Second, it addresses how such knowledge might be applicable in the realm of projects focused primarily on the development and implementation of IS. The topic represents the confluence of three streams of thought pertaining to strategy, IS projects, and project management broadly. We will scan each of these literature streams for reference to the implementation of strategy through IS projects, to note linkages or missing links in the extant knowledge base, and to use these insights to suggest a theoretical model in the form of propositions to guide future research.

We follow Webster and Watson’s (2002) guidance by first describing the key concepts and delineating the boundaries of our research, along with a set of questions driving our investigation. Next, we describe our methods, followed by a short example of how the analysis might unfold. Moving from this RIP to the full paper, this will be followed by a model and propositions to guide future research. Our process is congruent with Vom Brocke and colleagues’ description of how to conduct an MIS review. Our objective is explanation building through a theoretical review. Our scope is broad, and sources of material will include both conceptual and empirical research. Further, we assume knowledge exists regarding organizational strategy and its enactment in general, with execution through projects as one approach. We assume knowledge exists about better ways to implement IT-enabled projects, although not necessarily ones that aim to maximize strategic goals. In sum, we will review the following literatures: project management in terms of enacting strategic purposes, strategy pertaining to implementation, and IT implementation and IT project management. We will also evaluate the enterprise project governance literature (e.g., Dinsmore and Rocha, 2012), which integrates organizational strategy and effective project conduct.

Organizational strategy will be considered mainly from a top-down formal strategy planning perspective. This is a dominant perspective on strategy and is most likely to initiate a strategic push toward the design and execution of IS projects. Some scholars view strategy as emergent, most notably Mintzberg (1996), and we will examine issues of de facto strategy that may be emergent from IS project processes and deliverables.

Projects are activities with a beginning and end aimed at creating particular products and/or services that did not exist before. They vary on many dimensions including size, the nature of the product/service, their complexity, the membership and structure of teams, the work packages, the risks, and the stakeholder characteristics (Project Management Institute, 2013). More specifically, IT-enabled projects are focused activities with a beginning and end aimed at creating, deploying, changing, organizing, or otherwise structuring the computing and information resources of an organization. Such projects may address the range of levels of the technology stack from platform to content, with software applications somewhere in between. IS projects are generally of a socio-technical nature, implying some adjustment to the technical state or tools available within an organization as well as to the skills, approaches,
assignments, and understandings of the humans interacting with this technology. Thus, project management “is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements” (Project Management Institute, 2013).

We bound our investigation by only looking at details of IS project implementation as they apply to the enactment of strategy. Among the distinctions we make are to focus on strategic initiatives and projects, rather than compliance or emergency project development (e.g., Larson and Gray, 2014). Issues of strategic alignment of organizations and IS departments are largely be outside the boundary of this study, other than where they suggest or discuss projects per se as a mechanism for integrating alignment. General knowledge about how to run effective IS projects, such as critical success factors (CSFs) for agile or traditional approaches will be outside the scope of the study other than where such CSFs affect the ability to implement strategic goals.

Guiding Questions

The following questions guide our investigation and theory building.

- What is known about how IT-enabled projects can be used to enact organizational strategy?
- What is missing in the general project management, IS implementation, and organizational strategy literatures relative to IS project execution and strategy realization?
- Are there processes, factors, techniques, or approaches that differentiate more from less successful enactment of firm strategy through the use of IS projects?
- Do IT-enabled projects differ from other projects relative to more effective strategy execution?
- Are there macro processes for handling the class of IS projects or micro processes for actions within IS projects that make them more effective at enacting organizational strategy?
- Can we integrate different directions of influence: how IS project management capabilities constrain the range of strategic foci for an organization as well as how organizational strategy can influence the organization and execution of IS projects toward manifesting strategic vision?

METHODS

Because there is little research directly targeting the mechanics of implementing strategy through projects and programs, our review will examine three separate literature streams: organizational strategy, IS projects, and project management. Each of these offers the possibility of presenting direct observations of the interaction between organizational strategies and IS projects. We will search various digital libraries, most notably ABI Inform, ACM digital library and Business Source Premier, based on key words focused on all three topics and each pair of topics. In anticipation of large numbers of “hits”, we will select articles from 3-10 top journals in each of these literature streams. For example, in MIS, we will select articles from the eight journals in the senior scholars’ basket, and Information and Organization and Information and Management, which both have a significant number of manuscripts that address organizational level IS issues. In project management, we will select articles from the discipline’s three top journals - International Journal of Project Management, Project Management Journal, and International Journal of Managing Projects in Business. We will make a similar selection for strategy including Organization Science, Academy of Management Journal, Academy of Management Review, Journal of Management, Journal of Strategic Management, Management Science, and Strategic Management Journal.

We will extend our search using a snowball technique by reviewing the reference lists in the articles we examine for additional relevant articles. Finally, we will use ABI Inform and/or Google Scholar to identify papers that have cited seminal articles within the boundary of our review to find less obvious but important references to our research question. We expect the vast majority of citations of these key articles to not offer much new information regarding our research question, but consider them worthy of scanning for the sake of completeness.
Analysis

Each paper is reviewed for information relevant to our research question. Where such relevant information is found we (1) extract those ideas, findings, definitions, and/or comments that apply to our research question; (2) qualitatively use techniques as described by Miles and Huberman (1994) and Corbin and Strauss (2014) to compare and contrast, to categorize, to quantify (where applicable), and to find patterns and observe the absence of extant knowledge; and (3) integrate these observations into a theoretical model and generate new propositions to guide future research. Specifically, we rely on abductive logic to sort and order the extracted content of these papers into a coherent pattern to facilitate the organization of related knowledge. We use a number of recommended theory building techniques including problematization (Alvesson and Sandberg, 2011) and contrastive explanation (Tsang and Ellsaesser, 2011). Analysis is supplemented with tactics of memoing, discussion to consensus, and theoretical saturation relative to a stopping rule for consideration of additional literature. We will provide a chain of evidence to illustrate how we triangulated and moved from an analysis of articles to conclusions. The objective is to find a categorization scheme that holds value as a way to organize the literature on this topic and, thereby, create theory based on the collected observations to date from the field. We have started with the project management literature as it seemed the most likely place to find answers already extant to our research question, but early examination shows that while this is a sometimes discussed phenomenon, there is little development of a detailed theoretical account for how this linkage is created and maintained.

Level of Analysis

The methodological level of analysis is the article or study. We examine each of these with a primarily qualitative approach, although some quantification may meaningfully arise from the analysis. We expect that the content of the articles may span various levels of analysis. Some literature may link strategy with organizational outcomes at a firm level. Other literature may link particular IS projects with strategic alignment. We do not anticipate research that integrates project and firm level results, but hold open the possibility. Any theoretical model we present will address this issue and discuss associated measurement issues (Klein and Kozloski, 2000).

INITIAL ANALYSIS

We have generated a set of 27 papers in the project management domain, published in the three targeted premier journals in the field. These are included in the Reference List. Initial analysis by one author has revealed (1) that no identified project management journal articles directly address the interaction of IS project management and strategy implementation in a thorough, theory-oriented, and compelling manner and that (2) the accumulated knowledge about project management, IS projects, and strategy execution present tantalizing glimpses and possibilities for extrapolation and theory building.

Our preliminary analysis surfaced two important distinctions or dimensions that need to be specified to understand the contribution of individual studies. These are the level of aggregation of projects and the actions, policies, and plans for varied project stages. We anticipate finding additional dimensions when examining further literature and will consider issues such as long term and short term effects, effects of strategy types and/or quality. For example, a poorly formulated strategy may be more difficult to guide even well run IS projects. Another promising avenue may derive from the directionality of influence: how different strategies affect the design of IS projects versus how variation in the execution of IS projects may enable or constrain strategy implementation at a firm level.

Temporality is another interesting dimension. Targeting variations based on project phase, for example, we can see that the work packages associated with a particular project may vary in their independence. Where they are highly independent, they may be performed in any order. Where they are highly interdependent they must be performed in a particular order. At a detailed work level, a program must be coded before it can be tested, and tested before debugged. However, for most projects at least a moderate level of interdependence creates a fairly regular series of types of work package execution in a particular order. These can be viewed in terms of (1) project planning and initiation, (2) project execution, and (3) project follow up. In terms of executing the organization’s strategic plans through project activity, each of these stages holds risks and opportunities.
The first stage involves what Pinto (2015, p. 13) calls conceptualization and planning. We combine the two in considering the first stage as all activities prior to the beginning of execution. The line between conceptualization and planning in practice is a thin one particularly considering how much variation there is in actual implementation of projects versus normative description of key activities. An important portion of project planning involves project selection: (1) strategies for selection; (2) goal and objective selection of a larger scale than individual projects (e.g. how selection can add up to a set of projects that through synergy provide benefits beyond the sum of the products of each project singly); and (3) consideration of particular projects with go-no-go decisions and where they may be along the continuum of high to zero alignment with strategic aims. Strategy operates relative to project planning in two simultaneous ways. First, it implicitly or explicitly suggests an alignment of the project’s goals and activities with those of the larger organization. Second, it suggests internal strategies for approaching the mechanisms by which the project operates to retain efficiency while effectuating its goals.

The second stage addresses project execution. A stream of literature pertaining to global project management suggests that communication about overall global strategy is one of four keys to global project success (Aarseth, Rolstadås, and Andersen, 2014). This study distinguishes between a global strategy and a global project strategy. The former refers to a global business strategy, while the latter refers to the local adjustments needed to enact that overall strategy (such that projects adjust to local culture, preferences, labor, government, regulations, yet still support the overall business strategy). The authors advocate relationship management approaches toward customizing projects, or portions of large global ones, to local environments. Three particular activities are promoted – identifying key local stakeholders (who may have very different attitudes from one location to another), develop global human resource management policies, and define global systems such as decision making, communication and reporting structures as well as information and format standards.

The third stage involves project follow up. This may include change management whereby the organization changes its personnel, structure, and/or processes in order to take advantage of the new capabilities created through the project. This is typically a fluid interplay where the organization changes and the project products may also be adjusted.

Moving forward, we will methodically examine the three literature streams. The second author will review these already identified articles for verification and extraction of additional content; both authors will extend this review to IS and organizational strategy literatures.

CONTRIBUTION AND CONCLUSION

This literature review will examine what has been learned so far about mechanisms designed to implement strategy through IS projects and whether there is evidence that these techniques have been effective. It addresses how such knowledge might be applicable specifically in the realm of projects focused primarily on the development and implementation of information technology. It is clear that (1) a very high percentage of projects, particularly those creating new products and/or services internally or externally to firm involve in part or whole the development or deployment of information technologies; (2) that by their nature much of the process of developing IS is of an intangible nature and functions differently (e.g. stakeholders, risks, schedule, and assessing partial completion of tasks) from projects; and (3) that IS projects present a very rich target where understanding, predictability, and helpfulness can have a significant positive impact on practice. In summary, it is always difficult to be precise on what will be found with a work in process. Initial searching of the literature shows: (1) that indeed this is a topic not explicitly addressed often; (2) that generates new and interesting questions; and (3) that promises new insights through the arranging of and extrapolation among extant materials.

Both IS and strategy are part of the central core of interest for the SIGITPROJMGMT community. This study will illuminate the connection between project management, an organization’s strategy, and specifically how IS projects may or may not instantiate them. The topic, therefore, is a formidable complement to the study of systems and how they lead to advancing organizational strategies. By examining mechanisms for executing strategy, findings should be relevant both to academics in terms of providing insights for further testing and/or refinement as well as practice for forming a basis for predicting outcomes and selection of execution process techniques. The study pertains to consideration of a topic not well covered but one that appears on “the shop floor” consistently in practice. We are focused on theory building but also on providing utilitarian understanding of the phenomenon (Galliers, Jarvenpaa, Chan, and Lyytinen, 2012). We believe implications of the study’s findings will be relevant broadly to those interested...
in project management generally, IS management, and in strategy, particularly in terms of its execution. We fully expect that future work will include the potential for (1) empirically testing the propositions we put forth and (2) further exploring parallel or even conflicting alternative approaches to executing strategy through IS projects.

In project management, a growing area of importance pertains to benefits realization: techniques to achieve on time and on budget efficiency (Ashurst et al., 2008). However, in the broader sense it refers to aligning efforts with strategic purpose and realizing the purposes for which the project was selected and gaining benefits from its effectiveness. How strategy is realized through IS projects represents a timely and important area of study. Our contributions are to (1) collect what is known about this phenomenon in current literature; (2) emphasize the mechanisms and processes by which such strategic benefits are realized; and (3) consider specifically the role of IS and both behavioral and technical issues that may impact the effectiveness of particular mechanisms.

REFERENCES


Project Management Institute (2013). A guide to the project management body of knowledge. 5th Ed., Newtown Square, PA, USA.


**Results of Search on Key Words - ABI Inform May 4, 2017**


**Identified but not yet Evaluated Relevant Articles by “Snowball” Consideration of Citations of Key Articles**


Temporality in Information Systems Development (ISD) Research: A Systematic Literature Review

Mairead O Connor
Lero - The Irish Software Research Centre, National University of Ireland, Galway
m.oconnor67@nuigalway.ie

Kieran Conboy
Lero - The Irish Software Research Centre, National University of Ireland, Galway
kieran.conboy@nuigalway.ie

Denis Dennehy
Lero - The Irish Software Research Centre, National University of Ireland, Galway
denis.dennehy@nuigalway.ie

ABSTRACT

While time is central to the way we work and live our lives, it is often viewed overly simply or overlooked completely. Time is highly complex, polymorphous, socially constructed and context dependent. Time is also experienced differently across cultures, subcultures, organisations, teams and individuals. Many researchers have attempted to classify the complexities of time. However, there is no overarching framework which is commonly agreed upon. We reviewed the temporality literature and particularly the existing frameworks which attempt to classify time. This research-in-progress paper uses the well regarded Ancona et al., (2001) framework to classify time into three categories: conceptions of time, mapping activities to time and actors relating to time. Guided by this framework, the extant literature on information systems development (ISD) research was reviewed. Preliminary findings were found, conclusions, limitations and future research is also considered.

Keywords

Time, temporality, information systems development, systematic literature review.

INTRODUCTION

No matter the topic, or the context, time is always a prominent, multifaceted and complex concept (Ancona, Okhuysen & Perlow, 2001). Information systems development (ISD) is no exception. In fact, ISD usually takes place in a relentlessly dynamic, complex and unpredictable environment (Abdel-Hamid & Madnick, 1990; Benbya & McKelvey, 2006; Cockburn & Highsmith, 2001; Conboy, 2009). In such an environment, time becomes even more critically important (Nanadhakumar, 2002).

Contemporary and emergent ISD literature is filled with temporal characteristics but neglect the complex and subtle nature of time. Time as a construct is related to other constructs for example, in ISD literature pace (Vidgen & Wang, 2009), rhythm (Sarker & Sahay, 2004), velocity (Power & Conboy, 2015), speed (Conboy, 2009), just-in-time (Lee, 1999), on-time delivery (Sarker, Munson, Sarker, & Chakraborty, 2009), completion rates (Yetton, Martin, Sharma, & Johnston, 2000), lead time (Patmakunni & Ruppel, 2006), time management (Nanadhakumar, 2002), planning (Shmueli Pliskin, & Fink, 2016), time pressure (Austin, 2001) rapid change (Ramesh, Cao, & Baskerville, 2010), temporal dissonance (Conway & Limayem, 2011), and late deliveries (Austin, 2001) are all illustrations of temporal terms in ISD literature.

Yet, those studying and experiencing ISD conceive time as clock time. This over reliance on clock time overshadows ISD perspectives and theories (Nanadhakumar, 2002). As a result, traditional time management techniques are ill prepared for ISD projects (Nanadhakumar, 2002). This is illustrated by frequent late deliveries of ISD projects with continuous and problematic failure rates (Bartis & Mitev, 2008; Carlton, 2014; Pervan, 1998; Whittaker, 1999). In contemporary and emergent ISD method literature, time has not been studied as comprehensively as it should. Yet, time is the single differentiating factor between ISD methods to date (i.e, speed, sprints, daily stand ups, cycle time, lead time, velocity).
Considering the importance of time in ISD and following numerous calls to research temporality within the context of ISD teams and under dynamic conditions, it has not been explored explicitly or sufficiently in information systems (IS) research (Lee & Liebenau 2000; Saunders & Kim 2007; Waller, Conte, Gibson, & Carpenter, 2001).

Building on the comprehensive temporality framework proposed by Ancona et al., (2001), the aim of this research is to understand the various components of temporality within the context of ISD. For example, clock time is placed in such importance in all ISD projects, but an important question to ask is whether clock time is the best way to conceptualise time in ISD projects. Perhaps other categories of time would improve ISD project success. This research will examine the concept of temporality within ISD teams and produce a systematic literature review which will give the reader an up-to-date analysis of temporality in ISD. This research contributes to research and practice by identifying any gaps, misconceptions or general conceptual issues in the application of temporal concepts to ISD.

This research-in-progress paper describes the research objective, background literature, concept of temporality, research approach to examining time in ISD literature, and preliminary literature review findings.

RESEARCH OBJECTIVE

We aim to address the call from ISD researches which found: (i) time is understudied; (ii) time is dominated by clock time and neglects all other conceptions of time; (iii) ISD is inherently characterised by time but studies which address time explicitly is lacking.

A fundamental motivation for this research is: (i) frequent late deliveries of ISD projects with continuous and problematic failure rates show little signs of improvement; (ii) traditional time management techniques are ill prepared for ISD projects; (iii) those studying and experiencing ISD conceive time as clock time; (iv) clock time overshadows ISD perspectives and theories; (v) time is the single differentiating factor between ISD methods to date (i.e, speed, sprints, daily stand ups, cycle time, lead time, velocity). Yet, in contemporary and emergent ISD method literature, time has not been studied as comprehensively (Ancona et al., 2001 framework) as it should; (vi) no literature review of time in ISD research has previously been published. Thus, we aim to address if the complexity of time is studied in ISD.

The objective of the review is to answer the following research questions:

**RQ1** What has been reported about temporality in ISD in the existing literature?

**RQ1.1** What component of temporality is being studied?

**RQ2** How is temporality studied in ISD?

**RQ2.1** What methodology is used to study temporality?

**RQ3** How has temporality research contributed to ISD literature?

**RQ 3.1** What was the contribution to temporality?

In addition to enhancing findings on the complexity of time within ISD, the review also aims to contribute to conducting information systems systematic literature review methodology. A further contribution will be that of a study which conducts a systematic literature review: (i) where the complexity and subtle nature of time is incorporated into a search strategy; (ii) to find relevant papers; (iii) which are then systematically analysed based on a conceptual framework.
BACKGROUND LITERATURE

Over the past twenty-five centuries, the complexities of time have been studied in many disciplines such as psychology (Friedman, 1990), physics (Einstein, 1905; Hawking, 1993), and anthropology (Durkheim, 1915). There has been a renewed interest in time and timing in organisational theory (Crossan, Cunha, Vera, & Cunha, 2005; Kamp, Lambrecht Lund, & Sondergaard Hvid, 2011; Roe, 2008). This can be attributed to the emergence of global businesses and the urgent nature of production demands (Orlikowski & Yates, 2004). In organisational studies, one of the first and most famous studies to explicitly concentrate on time was by Taylor (1911). Taylor revolutionised the way in which managers view time and much of his findings can still be seen across organisations today (Guerrier, 2016). However, somewhat remarkably, very little research exists on time in an organisational setting over the years (Ancona et al., 2001). Temporality studies should have been included in organisational theory; however, it has remained hidden for decades (Sonnentag, 2012). However, in recent years, organisational researchers have begun acknowledging the benefits of analysing organisations through a temporal lens (Ancona et al., 2001, Orlikowski & Yates, 2004; Sonnentag, 2012).

Time underpins much of what we think about and do in IS. This is also reflected by the interest of time and IS in organisational work (Saunders & Kim, 2007; Sarker & Sahay, 2004; Nandhukumar, 2002; Lee & Liebenau, 2000; Lee, 1999). IS researchers regularly stress the time benefits of IS within the organisation. However, IS researchers are slow to address this polymorphous, multi-dimensional, complex concept that is often masked subtlety and over-simply in IS research (Saunders & Kim, 2007; Nandhukumar, 2002). As a result, research on time in IS is largely under-explored (Lee & Liebenau, 2000) and there is a need for more research within the area (O’Riordan, Conboy & Acton, 2013; Saunders & Kim, 2007; Shen, Lytinen & Yoo, 2014). The one-dimensional focus of clock time in IS research means that temporal synonyms such as speed, pace, velocity, cost of delay, ignore the complexity around time which permeate IS research. A multi-dimensional view of time would consider the perception of time and an actor’s experience of time in IS teams (Shen et al., 2014). The need to produce temporality studies in IS which go beyond the simple, one dimensional view of linear clock based time is ongoing (O Riordan et al., 2013).

ISD is a time pressure and highly complex environment where regular delays occur (Austin, 2001). There are frequent innovative ISD methods introduced into the industry to solve delays. The rationale for this ever-emerging series of ISD methods is time. For example, faster time to market. The timing and sequence of activities and not the actual activities themselves have been to some degree, the differentiation between methods. It is not the activities but the length of time and sequence of those activities that differentiate various methods such as waterfall, rapid development, agile, lean, flow and continuous development from each other. For example, agile methods promote an ISD environment where customer requirements can be changed more quickly than traditional methods (Dingsøyr & Lindsjørn, 2013).

TEMPORALITY

Temporality refers to an individual’s experience of time (Caldas & Berterö 2012), which includes our relationship to time (Heidegger, 1927), and how we react to time (Fraisse, 1963). There have been explicit attempts to add to temporality theory within the past few decades (Ancona et al., 2001; Mosakowski & Earley, 2000; Orlikowski & Yates, 2002; Standifer & Bluedorn, 2006). However, there is a lack of an overarching temporal framework. There are several attempts to create an overarching theory on time, however no theory of time is universally accepted.

In organisational literature, several temporal frameworks have emerged over the last four decades, mostly within organisational research (e.g., Bluedorn & Denhardt, 1988, Ancona et al., 2001, Mosakowski & Earley, 2000, Orlikowski & Yates, 2002, Sonnentag, 2012). Scholars have used a range of temporal dimensions to describe temporality in organisations. There are frameworks (Bluedorn & Denhardt, 1988; Ancona et al., 2001) which consider the full extent of time and all its complexity. These studies not only consider all temporal dimensions but also categorise them into an applicable structure. The Bluedorn & Denhardt (1988) framework classifies time into three categories; social time, mathematical time, and economic time. Ancona et al., (2001) classifies time into three interrelated categories; conceptions of time, socially constructed time, and actors relating to time. However, a concept analysis reveals that there are inconsistencies with the temporality literature that exists. There are inconsistent definitions and use of temporal terms in these frameworks.
There are studies which examined temporal dimensions within a problem. These temporal dimensions vary. Schriber & Gutek (1987) examined temporality within organisational culture and found thirteen main temporal dimensions; time boundaries between work and non-work, sequencing of tasks, punctuality, allocation, awareness, synchronization and coordination, variety versus routine, intra-organizational time boundaries, future orientation, schedules and deadlines, work pace, autonomy of time use, and quality versus speed. Lee & Liebenau (2000) identified six temporal dimensions that can be used to evaluate the temporal effects of IS: duration, sequence, temporal location, deadline, cycle and rhythm. Zerubavel (1981) examined temporality in organisations and found four temporal dimensions: sequential structure, duration, temporal location and rate of recurrence. Ancona et al., (2001), Schriber & Gutek (1987), Lee & Liebenau (2000), and Zerubavel (1981) all examined temporality within an organisational context. Schriber & Gutek (1987), Lee & Liebenau (2000), and Zerubavel (1981) contributed to temporal dimensions. However, Ancona et al., (2001) framework is unique as it provides a holistic classification of temporal dimensions.

**Classification of time**

This study uses the Ancona et al., (2001) framework (Table 1) to firstly classify time and secondly to understand what ISD literature on time exists to date. Ancona et al., (2001) reviewed the literature surrounding temporality within the areas of organisational theory, sociology, social psychology, and anthropology and developed a framework to classify time. The framework has three categories called conceptions of time, mapping activities to time and actors relating to time (Ancona et al., 2001). The framework was chosen because it is a well-regarded classification of time (Shen et al., 2014). Shen et al., (2001) also chose it because it “synthesizes a large swath of temporal concepts across diverse areas of temporal study and provides a common organising framework for these temporal constructs and variables”. It is also an easily applied structure that can be used as a lens to examine teams (Shen et al., 2014, p3). Thus, it is suited to this research on ISD.

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Sample Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptions of time</td>
<td>Types of time</td>
<td>Linear time, uniform time, cyclical time, subjective time, event time</td>
</tr>
<tr>
<td></td>
<td>Socially constructed time</td>
<td>Work organization (nine-to-five workdays, work time and family time), celebrations (Passover and/or Easter), time as a renewing cycle, time as linear continuity</td>
</tr>
<tr>
<td>Mapping activities to time</td>
<td>Single activity mapping (a)</td>
<td>Scheduling, rate of completion, duration</td>
</tr>
<tr>
<td></td>
<td>Repeated activity mapping (aa)</td>
<td>Cycle, rhythm, frequency, interval</td>
</tr>
<tr>
<td></td>
<td>Single activity transformation mapping (aa’)</td>
<td>Life cycles, midpoint transitions, jolts, interrupts, deadline behavior</td>
</tr>
<tr>
<td></td>
<td>Multiple activity mapping (ab)</td>
<td>Relocation of activities, allocation of time, ordering, synchronization</td>
</tr>
<tr>
<td></td>
<td>Comparison and meshing of activity maps (ab) versus (aa)</td>
<td>Entrainment, patterning, temporal symmetry</td>
</tr>
<tr>
<td>Actors relating to time</td>
<td>Temporal perception</td>
<td>Experience of time, time passing, time dragging, experience of duration, experience of novelty</td>
</tr>
<tr>
<td></td>
<td>Temporal personality</td>
<td>Temporal orientation, temporal style</td>
</tr>
</tbody>
</table>

*Table 1. Classification of Categories and Subcategories, with Sample Variables (Ancona et al., 2001)*
Conception of Time

The framework identifies three categories; conceptions of time, mapping activities to time and actors relating to time. Each classification of time is now discussed in turn. A conception of time can be created by an individual. For example, an employee in an organisation can conceptualise their own view of time (Medlin & Saren, 2012). There are different types of time such as linear time, uniform time and cyclical time. However, the most popular and widely cited types of time are clock time and event time (Mosakowski & Earley, 2000). Each society will conceptualise time differently. For example, some may use clock-based time whereas others may use event-based time (Mosakowski & Earley, 2000). Clock time versus event time can be associated with the Newton (1871) Vs Einstein (1945) debate. Newton perceived time as absolute and Einstein perceived time as relative. Still, individuals can experience multiple types of time at once (Ancona et al., 2001). Time is also revealed as a socially constructed phenomenon (Ancona et al., 2001; Saunders et al., 2004). Even within these groups, time can be used in a way which is appropriate to the sub group (Ancona et al., 2001). This means there can be different dimensions of time within a society (Saunders et al., 2004).

Mapping activities to Time

The main aim of mapping activities to time is to get a valid analysis of what happens over time during an activity (Roe, 2008). This category explains that events and activities can be mapped to time. Mapping activities to time explains when an activity will begin, how long it may take and any fluctuations or patterns over the interval (Ancona et al., 2001). Single activity mapping maps a single activity to a time continuum (Ancona et al., 2001). Single activity mapping entails the examination of a single event which is not usually repeated once the activity has taken place. Repeated activity mapping has been researched far less than single activity mapping. Repeated activity mapping is where an activity is repeated on multiple occasions and is mapped to time. Single activity transformation mapping occurs when, during an activity, the old activity transforms into the new activity, changing the form of the activity (Ancona et al., 2001). The single activity transformation mapping is catered for one single event. In multiple activity mapping, activities are examined in relation to each other. The primary concern for multiple activity mapping is allocating the correct amount of time towards the multiple activities (Ancona et al., 2001). Comparing multiple activity mapping is used to understand the differences and similarities in temporal characteristics. Not all activities have the same temporal characteristics.

Actors relating to time

Actors relating to time explain the actors which are involved in the previous two categories. Temporal perception variables are used to reveal how actors perceive and act with regard to the continuum of time. The actors may refer to the individual, team or the organisation (Ancona et al., 2001). The perception of time is different among the different cultures around the world (Mosakowski & Earley, 2000). Individuals also have their own temporal personality (e.g., when an individual schedules his/her tasks). They will follow their own temporal personality while doing so (Avnet & Sellier, 2011). Although our perception of time is influenced by our culture, it is also formed by our own individual temporal personality. A temporal personality is unique to the distinctive individual and the way in which an individual understands and experiences time will be exposed through their actions (Ancona et al., 2001).

RESEARCH METHOD

General temporality and time classification frameworks were examined. Having chosen the Ancona et al., (2001) framework because of its broad application and easily applied structure, a number of pilot tests were carried out. We used a systematic literature review approach and a brief analysis of temporal research in ISD was of a primary concern. From that, some opening early stage preliminary findings were exposed.

In conducting our literature review, we chose an efficient scientific method; a systematic literature review. A systematic literature review is a “means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest” (Kitchenham, 2004:1). This type of review offers a high quality (Dybå & Dingsøyr, 2008), transparent and replicable review (Leidner & Kayworth, 2006). This method offers the capability of summarising a large quantity of research publications (Fink, 2005), for studies which aim to address a clearly formulated question (Petticrew & Roberts, 2006). Therefore, a systematic literature review was chosen rather than a narrative literature review for four reasons: (1) our study aimed to answer a specific research question; (2) our area of study will generate a large amount of literature; (3) our intention was to systematically extract...
relevant temporal references from the publications in a transparent form; (4) the rigour and replicability it offers which leads to an unbiased scientific paper.

Systematic literature reviews originated in health science (Fink, 2005) and have been applied in many diverse domains including supply chain management (Gimenez & Tachizawa, 2012), nursing (Brady Germain & Cummings, 2010), organisational management (Crossan & Apaydin, 2010), marketing (Antunes & Moreira, 2011) and air transport (Ginieis et al., 2012). Within information systems, systematic literature reviews have been published within the top journals (Grahmann, Helms, Hilhorst, Brinkkemper, and Van Amerongen, 2012; Lavranos, Kostagiolas, Korfiatis, & Papadatos, 2015). However, not to a large extent. Two reasons for this neglect has been that IS researchers are unaware of the need for a rigorous and systematic literature review (Levy & Ellis, 2006) and many are unsure about the methodology involved in conducting a systematic literature review (Okoli & Schabram, 2010).

Systematic literature review guides have also been developed in a variety of different domains such as social science (Petticrew & Roberts, 2006), software engineering (Kitchenham & Charters, 2007) and IS (Leidner & Kayworth, 2006; Levy & Ellis, 2006; Okoli & Schabram, 2010; Webster & Watson). IS research methods are different from health sciences methods and require a specific systematic literature review guide (Okoli & Schabram, 2010). Within IS, bespoke systematic literature reviews were developed in response to the shortage of appropriate guides which suit the multidiscipline domain of IS. The domain is made up of social science, business, and computing science disciplines (Okoli & Schabram, 2010).

We followed the systematic literature review approach based on work by Kitchenham & Charters (2007), Levy & Ellis (2006), Okoli & Schabram (2010) and Webster & Watson (2002). The foundation of our guide was taken from an IS guideline developed by Okoli & Schabram (2010). However, we drew from similar work by Kitchenham & Charters (2007), Levy & Ellis (2006) and Webster & Watson (2002) to strengthen our scientific review.

Webster & Watson (2002) was one of the first attempts to create a systematic literature review guide within the IS domain. This work has contributed to further systematic literature review guide within IS such as Levy & Ellis (2006) and Okoli & Schabram (2010). However, it lacks the depth Levy & Ellis (2006) and Okoli & Schabram (2010) can offer. Webster & Watson (2002) lacks rigour and explanation in which the novice researcher needs. Hence, Levy & Ellis (2006) and Okoli & Schabram (2010) were chosen to add strength and further thorough guidelines to the systematic literature review process.

Levy & Ellis (2006) developed a set of guidelines for IS doctoral researchers, early stage researchers and IS researchers who have difficulties completing a successful literature review. Levy & Ellis (2006) took some features from Webster & Watson (2002) such as the backward and forward search feature. However, it offers a more in-depth guide tailored for the needs of the IS doctoral researchers.

Okoli & Schabram (2010) developed a bespoke methodological approach for IS researchers which offers a guideline created from contributions within software engineering (Kitchenham & Charter, 2007), social sciences (Petticrew and Roberts, 2006), health sciences (Fink, 2005), management and organization science (Rousseau et al., 2008) IS (Levy & Ellis, 2006; Webster & Watson, 2002). This is an exhaustive guideline as it is adapted best practices from several other domains. It offers a balanced approach of both qualitative and quantitative methods. This differs to Levy & Ellis (2006) which is primarily qualitative based. This guide has more depth than Levy & Ellis (2006) and focuses on guidelines for areas that previous methodologies have neglected.

Kitchenham (2004) developed a lengthy and thorough procedure for performing literature reviews. These guidelines were progressed by Kitchenham & Charters (2007). Although Kitchenham & Charters (2007) is an extensive and exhaustive guide, it was created for the software engineering domain. However, software engineering overlaps with the IS domain and due to the guides large application within other fields such as business process management (Maita et al., 2005), marketing (McHugh & Domgan, 2010), and nursing (Beraldi & Abades, 2014), it has been chosen for the unique exhaustive contribution it can offer. One of its unique contribution is the protocol feature, in which we implemented for this paper.

We followed an eight-step guideline which is required for completion of a systematic literature review. Although any of these steps can be followed by researchers who complete a narrative literature review, following all steps are essential to conduct a scientifically rigorous systematic literature review (Okoli & Schabram, 2010). This guide is
based on (Okoli & Schabram, 2010) but draws from similar best practices from three other systematic literature review

A keyword strategy (Table 2) was developed based on the temporal terms used in several temporality frameworks
(Ancona et al., 2001; Bluedorn & Allen, 1988; Kavanagh & Araujo, 1995; Lee & Liebenua, 2000; Mosakowski &
Early, 2000; Sonnentag, 2012). They keyword strategy was applied to Association for Information Systems (AIS)
senior scholars’ basket of IS journals and top two IS conferences (Table 3). The largest, most extensive and
information systems-specific databases were chosen (Table 4). The research questions (Table 5) were used to develop
the inclusion (Table 6) and exclusion (Table 7) strategy. The data extraction (Table 8) and quality assessment (Table
9) was used to develop preliminary findings.

### Table 2. Keyword Strategy

| TOPIC: (( system* OR software* )) AND TOPIC: (( method* OR design* OR develop* OR practice* OR project*
OR agile* OR scrum* OR "extreme programming" OR xp OR feature* OR orystal* OR lean* OR flow* OR team*
OR technique* OR approach* )) AND TOPIC: ((tim* OR temporal* OR *cyc* OR schedul* OR rate* OR duration*
OR rhythm* OR frequen* OR interval* OR jolt* OR interrupt* OR deadline* OR *sync* OR order* OR pattern*
OR entrain* OR novel* OR "relocation of activities" OR "mid-point* transition*" OR "midpoint* transition*"
OR clock* OR calendar* OR speed* OR slow* OR "prior* OR routine* OR fast* OR year* OR event* OR day* OR
plan* OR monochronic* OR polychronic* OR pace* OR lag* OR urgen* OR sequen*))

### Table 3. Association for Information Systems (AIS) Senior Scholars’ Basket of IS Journals and Top Two IS Conferences

<table>
<thead>
<tr>
<th>No.</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Association for Information Systems</td>
</tr>
<tr>
<td>(2)</td>
<td>Scopus</td>
</tr>
<tr>
<td>(3)</td>
<td>Web of Science</td>
</tr>
</tbody>
</table>

### Table 4. Literature Databases

<table>
<thead>
<tr>
<th>ID</th>
<th>Research Question</th>
<th>Aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>What has been reported about temporality in ISD in the existing literature?</td>
<td>To provide a comprehensive review of time in ISD literature</td>
</tr>
<tr>
<td>RQ1.1</td>
<td>What component of temporality is being studied?</td>
<td>To establish what components of time are being studied within ISD literature</td>
</tr>
<tr>
<td>RQ2</td>
<td>How is temporality studied in ISD?</td>
<td>To identify how research on time in ISD is carried out</td>
</tr>
<tr>
<td>RQ2.1</td>
<td>What methodology is used to study temporality?</td>
<td>To establish what methodology is used in carrying out research on time in ISD literature</td>
</tr>
<tr>
<td>RQ3</td>
<td>How has temporality research contributed to ISD literature?</td>
<td>To identify the contribution temporality has made to ISD literature</td>
</tr>
<tr>
<td>RQ 3.1</td>
<td>What was the contribution to temporality?</td>
<td>To categorise the contribution of temporality in ISD literature</td>
</tr>
</tbody>
</table>
### Table 6. Inclusion Strategy

<table>
<thead>
<tr>
<th>No.</th>
<th>Inclusion strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Each study should relate to one or more of our research questions</td>
</tr>
<tr>
<td>(2)</td>
<td>Each study should fall into the top eight IS journals and the top two IS conferences</td>
</tr>
<tr>
<td>(3)</td>
<td>Each study should focus on temporality in ISD</td>
</tr>
<tr>
<td>(4)</td>
<td>Each study should be empirical, theoretical, conceptual, literature review or experimental</td>
</tr>
<tr>
<td>(5)</td>
<td>Each study should be published between 2000 and 2016</td>
</tr>
<tr>
<td>(6)</td>
<td>If the study has been published in more than one journal or conference, the most recent version of the study is included</td>
</tr>
</tbody>
</table>

### Table 7. Exclusion Strategy

<table>
<thead>
<tr>
<th>No.</th>
<th>Exclusion strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Duplicate articles will be excluded</td>
</tr>
<tr>
<td>(2)</td>
<td>Papers not written in English will be excluded</td>
</tr>
<tr>
<td>(3)</td>
<td>Lesson learned, research in progress, editor’s reports and experience reports will be excluded</td>
</tr>
<tr>
<td>(4)</td>
<td>Papers which use students as the sample study will be excluded</td>
</tr>
</tbody>
</table>

### Table 8. Data Extraction

<table>
<thead>
<tr>
<th>No.</th>
<th>Data extraction</th>
<th>Related RQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>What component of temporality is being studied?</td>
<td>RQ1</td>
</tr>
<tr>
<td>(2)</td>
<td>What methodology was used to study temporality?</td>
<td>RQ2</td>
</tr>
<tr>
<td>(3)</td>
<td>What was the contribution to temporality?</td>
<td>RQ3</td>
</tr>
<tr>
<td>(4)</td>
<td>Research setting</td>
<td>Overview</td>
</tr>
<tr>
<td>(5)</td>
<td>Article title</td>
<td>Overview</td>
</tr>
<tr>
<td>(6)</td>
<td>Author</td>
<td>Overview</td>
</tr>
<tr>
<td>(7)</td>
<td>Year</td>
<td>Overview</td>
</tr>
<tr>
<td>(8)</td>
<td>What ISD methodology does this cover?</td>
<td>Overview</td>
</tr>
<tr>
<td>(9)</td>
<td>Classification of study type</td>
<td>Overview</td>
</tr>
<tr>
<td>(10)</td>
<td>Source</td>
<td>Overview</td>
</tr>
</tbody>
</table>

### Table 9. Quality Assessment questions (source: Dybå and Dingsøyr, 2008)

<table>
<thead>
<tr>
<th>ID</th>
<th>Quality Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Is this a research paper?</td>
</tr>
<tr>
<td>Q2</td>
<td>Is there a clear statement of the aims of the research?</td>
</tr>
<tr>
<td>Q3</td>
<td>Is there an adequate description of the context in which the research was carried out?</td>
</tr>
<tr>
<td>Q4</td>
<td>Was the research design appropriate to address the aims of the research?</td>
</tr>
<tr>
<td>Q5</td>
<td>Was the recruitment strategy appropriate to the aims of the research?</td>
</tr>
<tr>
<td>Q6</td>
<td>Was there a control group with which to compare treatments?</td>
</tr>
<tr>
<td>Q7</td>
<td>Was the data collected in a way that addressed the research issue?</td>
</tr>
<tr>
<td>Q8</td>
<td>Was the data analysis sufficiently rigorous?</td>
</tr>
<tr>
<td>Q9</td>
<td>Has the relationship between researcher and participants been considered to an adequate degree?</td>
</tr>
</tbody>
</table>
PRELIMINARY FINDINGS

While the systematic literature review has not been fully conducted to date, some notable findings have emerged. Each finding is now discussed in turn. Firstly, preliminary findings associated with general temporality are summarised. Then, preliminary findings associated with each category of time is discussed in Table 10.

<table>
<thead>
<tr>
<th>Preliminary Findings Associated with General Temporality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time in ISD is studied implicitly rather than explicitly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preliminary Findings Associated with Conceptions of Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>One dimensional view of the concept of time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preliminary Findings Associated with Mapping Activities to Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of research involving mapping activities to time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preliminary Findings Associated with Actors Relating to Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of research involving actors relating to time</td>
</tr>
</tbody>
</table>

Table 10. Preliminary Findings
CONCLUSION AND FUTURE RESEARCH

The literature review to date shows that ISD research: (i) lacks explicit studies on time; (ii) adopts an overly simplistic and one-dimensional view of time; (iii) only focuses on single activities e.g., a deadline and neglects the fact that teams will be involved in multiple tasks and projects at once; and (iv) lacked studies which considered the actors relating to time in ISD. The next research phase will include a more in-depth literature review on time in ISD. This upcoming literature review will expose a fuller, truer and deeper analysis of ISD studies on time. The review will firstly include an analysis of explicit studies on ISD time where temporal constructs are present and are the focus of study. Secondly, the review will assess ISD research where time is implicit and subtly referenced. Findings will be expanded and classified into each temporal category and within each subcategory. A limitation of the study is that our systematic literature review does not include journal and conference papers which were outside of our sampling criteria. The full systematic literature review aims to: (i) identify any misconceptions or general conceptual issues in the application of temporal concepts to ISD to date.

ACKNOWLEDGMENTS

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

REFERENCES

Fitzgerald, B., Hartnett, G. and Conboy, K., 2006. Customising agile methods to software practices at Intel
decision making in managing global software development teams. Information Systems Management, 27(3),
pp.247-252.
Gimenez, C and Tachizawa, M. (2012),"Extending sustainability to suppliers: a systematic literature review", Supply
Chain Management: An International Journal, Vol. 17 Iss 5 pp. 531 – 543
Ginieis, M., Sánchez-Rebull, M.V. and Campa-Planas, F., 2012. The academic journal literature on air transport:
Analysis using systematic literature review methodology. Journal of Air Transport Management, 19, pp.31-35.
Guerrier, Y., 2016. 9 ModernDay ‘Schmidts’: The Legacy of Taylorism in Elite ‘Professional’ Roles. Re-Tayloring
Management: Scientific Management a Century On, p.155.
Information Technologies (5:2), pp. 103-121.
Kitchenham, B., Charters, S.M., 2007 Guidelines for performing systematic literature reviews in software engineering.
systematic literature review. Journal of the Association for Information Science and Technology.
Leidner, D.E. and Kayworth, T., 2006. A review of culture in information systems research: Toward a theory of
information technology culture conflict. MIS quarterly, 30(2), pp.357-399.
Levy, Y. and Ellis, T.J., 2006. A systems approach to conduct an effective literature review in support of information
systems research. Informing Science, 9.
Maglyas, A., Nikula, U. and Smolander, K., 2011, August. What do we know about software product management?-a
systematic mapping study. In Software Product Management (IWSPM), 2011 Fifth International Workshop on
(pp. 26-35). IEEE.
Mainela, T., Puhakka, V. and Servais, P., 2014. The concept of international opportunity in international
entrepreneurship: a review and a research agenda. International Journal of Management Reviews, 16(1),
p.105-129.
McHugh, P. and Domegan, C., 2010. Systematic Reviews: Their Emerging Role in the Co-Creation of Social


Okoli, C. and Schabram, K., 2010. A guide to conducting a systematic literature review of information systems research.


