12-9-2017

The Role of Time Pressure in Software Projects: A Literature Review and Research Agenda

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

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 (Malgonde, 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 fulfils 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.

	Sequential Development	**Agile Development**
Requirements	presumed to be stable	expected to change
Process	sequential	iterative
Activities (e.g., design, implementation)	one-time	in each iteration
Customer Involvement	mainly during requirements engineering and testing	in each iteration
Documentation	extensive	limited, tailored to the project

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, IEEEXplore, ProQuest and ScienceDirect. For our search, we relied on a query within the search fields *title*, *abstract*, and *keywords* for each

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 constraint*") AND (program* OR software OR project* OR "information technology" OR "information system*")*

As time pressure is not new 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[1] (AIS). Additionally, we considered selected conferences[2] 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 (Malgonde 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.

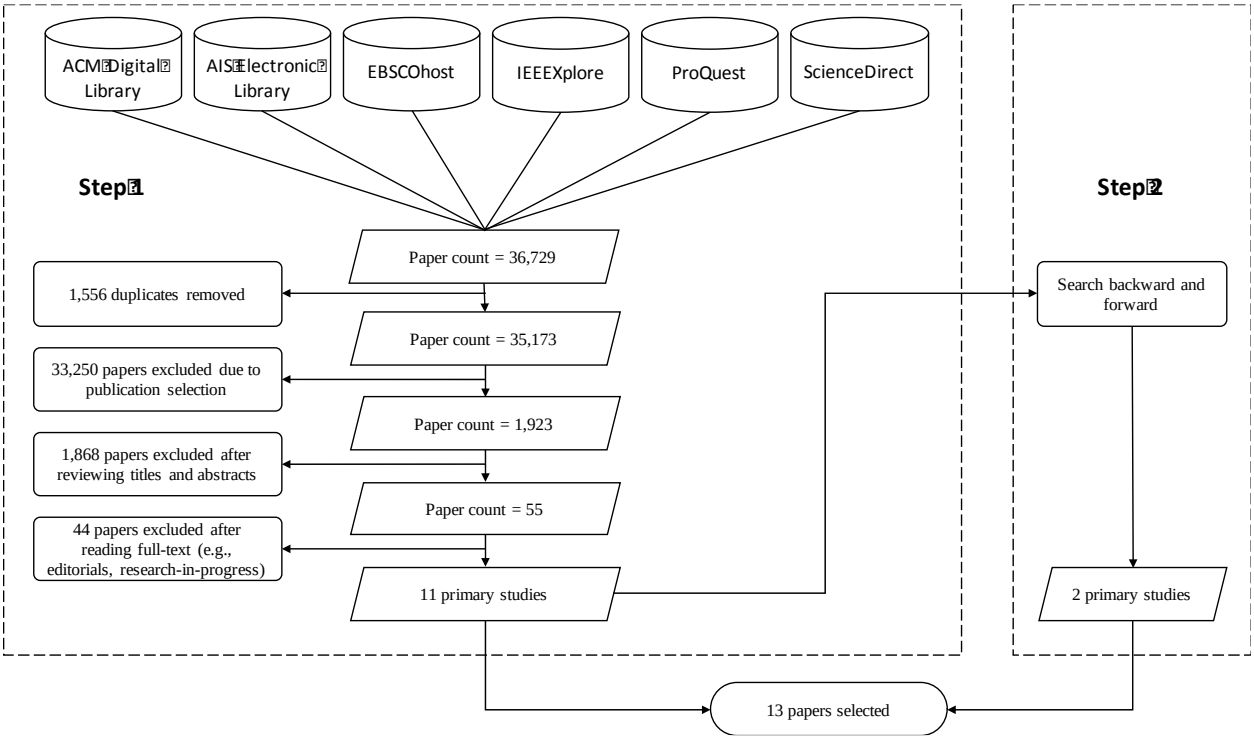

Figure 1. Selection of Publications

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

[1] Recently, the AIS removed the journal ranking from its website. It is available at: https://web.archive.org/web/20161007113451/https://aisnet.org/?JournalRankings

[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).

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.

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

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).

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

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

Afzal, W., Torkar, R. and Feldt, R. (2009) A Systematic Review of Search-based Testing for Non-functional System Properties, *Information and Software Technology,* 51, 6, 957-976.

Ågerfalk, P. and Fitzgerald, B. (2006) Flexible and Distributed Software Processes: Old Petunias in new Bowls?, *Communications of the ACM,* 49, 101, 27-34.

Asdemir, K., Kumar, N. and Jacob, V. S. (2012) Pricing Models for Online Advertising: CPM vs. CPC, *Information Systems Research,* 23, 3-part-1, 804–822.

Atkinson, R. (1999) Project Management: Cost, Time and Quality, Two Best Guesses and a Phenomenon, Its Time to Accept other Success Criteria, *International Journal of Project Management,* 17, 6, 337–342.

Austin, R. D. (2001) The Effects of Time Pressure on Quality in Software Development: An Agency Model, *Information Systems Research,* 12, 2, 195–207.

Banker, R. D., Datar, S. M. and Kemerer, C. F. (1987) Factors Affecting Software Maintenance Productivity: an Exploratory Study, in Proceedings of the International Conference on Information Systems, Association for Information Systems, Pittsburgh, 160-175.

Basten, D. and Sunyaev, A. (2014) A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation, *Communications of the Association for Information Systems,* 34, Article 4, 51–86.

Beck, K. (2000) Extreme Programming Explained: Embrace Change, Addison-Wesley, Reading.

Boehm, B. (2002) Get Ready for Agile Methods, with Care, *IEEE Computer,* 35, 1, 64-69.

Bourque, P. and Fairley, R. E. (2014) Swebok. Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, Los Alamitos.

Braun, V. and Clarke, V. (2006) Using Thematic Analysis in Psychology, *Qualitative Research in Psychology,* 3, 2, 77-101.

Cao, L. and Ramesh, B. (2008) Agile Requirements Engineering Practices: An Empirical Study, *IEEE software,* 25, 1, 60-67.

Cooper, C., Dewe, P. and O'driscoll, M. (2001) Organizational Stress: A Review and Critique of Theory, Research, and Applications, Sage, Thousand Oaks.

Deak, A., Stålhane, T. and Sindre, G. (2016) Challenges and Strategies for Motivating Software Testing Personnel, *Information and Software Technology,* 73, May 2016, 1-15.

Demarco, T. (1982) Controlling Software Projects. Management, Measurement & Estimation, Prentice Hall, New York.

Dennis, A. R. and Valacich, J. S. (2014) A Replication Manifesto, *AIS Transactions on Replication Research,* 1, Article 1, 1-4.

Do, H., Mirarab, S., Tahvildari, L. and Rothermel, G. (2010) The Effects of Time Constraints on Test Case Prioritization: A series of Controlled Experiments, *IEEE Transactions on Software Engineering,* 36, 5, 593-617.

Dybå, T. and Dingsøyr, T. (2008) Empirical Studies of Agile Software Development: A Systematic Review, *Information and Software Technology,* 50, 9-10, 833–859.

Fisher, C. W., Chengalur-Smith, I. and Ballou, D. P. (2003) The Impact of Experience and Time on the Use of Data Quality Information in Decision Making, *Information Systems Research,* 14, 2, 170–188.

Flick, U. (2009) An Introduction to Qualitative Research, Sage, Los Angeles.

Gersick, C. (1988) Time and Transition in Work Teams: Toward a new Model of Group Development, *Academy of Management Journal,* 31, 1, 9-41.

Gevers, J., Rutte, C. and Van Eerde, W. (2006) Meeting Deadlines in Work Groups: Implicit and Explicit Mechanisms, *Applied Psychology: An International Review,* 55, 1, 52-72.

Glass, R. L. (2004) Anarchy and the Effects of Schedule Pressure, *IEEE Software,* 21, 5, 112.

Grimstad, S., Jørgensen, M. and Moløkken-Østvold, K. (2006) Software Effort Estimation Terminology: The Tower of Babel, *Information and Software Technology,* 48, 4, 302–310.

Hwang, M. I. (1994) Decision Making under Time Pressure: A Model for Information Systems Research, *Information & Management,* 27, 4, 197-203.

Kappelman, L. A., Mckeeman, R. and Zhang, L. (2006) Early Warning Signs of IT Project Failure: The Dominant Dozen, *Information Systems Management,* 23, 4, 31–36.

Kelly, J. and Mcgrath, J. (1985) Effects of Time Limits and Task Types on Task Performance and Interaction of Four-person Groups, *Journal of Personality and Social Psychology,* 49, 2, 395-407.

Kitchenham, B. (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report, Keele University.

Kitchenham, B. A., Mendes, E. and Travassos, G. H. (2007) Cross versus Within-company Cost Estimation Studies: A Systematic Review, *IEEE Transactions on Software Engineering*, 33, 5, 316-329.

Kocher, M. G. and Sutter, M. (2006) Time is Money - Time Pressure, Incentives, and the Quality of Decision-making, *Journal of Economic Behavior & Organization*, 61, 3, 375–392.

Levina, N. (2005) Collaborating on Multiparty Information Systems Development Projects: A Collective Reflection-in-Action View, *Information Systems Research*, 16, 2, 109–130.

Linberg, K. R. (1999) Software Developer Perceptions about Software Project Failure: A Case Study, *Journal of Systems and Software*, 49, 2-3, 177–192.

Lohan, G., Acton, T. and Conboy, K. (2014) An Investigation into Time Pressure, Group Cohesion and Decision Making in Software Development Groups, in Australasian Conference on Information Systems, December 8-10, Auckland,

Mahaney, R. C. and Lederer, A. L. (2003) Information Systems Project Management: An Agency Theory Interpretation, *Journal of Systems and Software*, 68, 1, 1–9.

Mahaney, R. C. and Lederer, A. L. (2006) Agency Theory Implications for Information Systems Project Management, in Fred Niederman & Thomas W. Ferratt (eds.) *IT Workers. Human Capital Issues in a Knowledge-based Environment*, Information Age Pub., Greenwich, 85–104.

Malgonde, O., Collins, R. and Hevner, A. (2014) Applying Emergent Outcome Controls to Mitigate Time Pressure in Agile Software Development, in *Proceedings of Americas Conference on Information Systems*, Association for Information Systems, Savannah, 1-7.

Mäntylä, M. V. and Itkonen, J. (2013) More Testers – The Effect of Crowd Size and Time Restriction in Software Testing, *Information and Software Technology*, 55, 6, 986–1003.

Mäntylä, M. V., Petersen, K., Lehtinen, T. O. A. and Lassenius, C. (2014) Time Pressure: A Controlled Experiment of Test Case Development and Requirements Review, in A. C. M. Special Interest Group on Software Engineering (ed.) *Proceedings of the 36th International Conference on Software Engineering*, ACM, New York, 83–94.

Marks, M. A., Mathieu, J. E. and Zaccaro, S. J. (2001) A Temporally Based Framework and Taxonomy of Team Processes, *Academy of Management Review*, 26, 3, 356-376.

Maruping, L., Venkatesh, V., Thatcher, S. and Patel, P. (2015) Folding under Pressure or Rising to the Occasion? Perceived Time Pressure and the Moderating Role of Team Temporal Leadership, *Academy of Management Journal*, 58, 5, 1313-1333.

Miles, M. B. and Huberman, M. (1994) Qualitative Data Analysis: An Expanded Sourcebook, SAGE Publications, Thousand Oaks, California.

Mukhopadhyay, T., Vicinanza, S. S. and Prietula, M. J. (1992) Examining the Feasibility of a Case-based Reasoning Model for Software Effort Estimation, *MIS Quarterly*, 16, 2, 155–171.

Nan, N. and Harter, D. E. (2009) Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort, *IEEE Transactions on Software Engineering*, 35, 5, 624–637.

Niederman, F. and March, S. (2015) Reflections on Replications, *AIS Transactions on Replication Research*, 1, Article 7, 1-16.

Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T. and Ozkaya, I. (2013) Parallel Worlds: Agile and Waterfall Differences and Similarities, Carnegie Mellon University, Technical Report.

Pankratz, O. and Basten, D. (2015) One Size Does Not Fit All – Contingency Approach on Relevance of IS Project Success Dimensions, in Hawaii International Conference on System Sciences January, 5-8, Kauai, IEEE Computer Society, 4416-4425.

Pennington, R. and Tuttle, B. (2007) The Effects of Information Overload on Software Project Risk Assessment, *Decision Sciences*, 38, 3, 489-526.

Pinto, J. K. (2004) The Elements of Project Success, in David I. Cleland (ed.) *Field Guide to Project Management*, Wiley, Hoboken, 14–27.

Project Management Institute (2013) A Guide to the Project Management Body of Knowledge (PMBOK Guide), Project Management Institute, Newton Square.

Schwaber, K. and Beedle, M. (2002) Agile Software Development with Scrum, Prentice Hall, Upper Saddle River.

Shah, H., Harrold, M. J. and Sinha, S. (2014) Global Software Testing under Deadline Pressure: Vendor-side Experiences, *Information and Software Technology*, 56, 1, 6–19.

Shao, B. B. M., Yin, P.-Y. and Chen, A. N. K. (2014) Organizing Knowledge Workforce for Specified Iterative Software Development Tasks, *Decision Support Systems*, 59, 15–27.

Siau, K., Long, Y. and Ling, M. (2010) Toward a Unified Model of Information Systems Development Success, *Journal of Database Management*, 21, 1, 80–101.

Sojer, M., Alexy, O., Kleinknecht, S. and Henkel, J. (2014) Understanding the Drivers of Unethical Programming Behavior: The Inappropriate Reuse of Internet-accessible Code, *Journal of Management Information Systems*, 31, 3, 287-325.

Topi, H., Valacich, J. S. and Hoffer, J. A. (2005) The Effects of Task Complexity and Time Availability Limitations on Human Performance in Database Query Tasks, *International Journal of Human-Computer Studies*, 62, 3, 349–379.

Waller, M., Zellmer-Bruhn, M. and Giambatista, R. (2002) Watching the Clock: Group Pacing Behavior Under Dynamic Deadlines, *Academy of Management Journal,* 45, 5, 1046-1055.

Wang, E. T. G., Ju, P.-H., Jiang, J. J. and Klein, G. (2008) The Effects of Change Control and Management Review on Software Flexibility and Project Performance, *Information & Management*, 45, 7, 438–443.

Webster, J. and Watson, R. T. (2002) Analyzing the Past to Prepare for the Future: Writing a Literature Review, *MIS Quarterly*, 26, 2, 13-23.

Williams, L. and Cockburn, A. (2003) Agile Software Development: It's About Feedback and Change, *IEEE Computer*, 36, 6, 39-43.