

What Do We Know about Time Pressure in Software Development?

Miikka Kuutila, Mika Mäntylä, Umar Farooq, Maëlick Claes

Abstract—Time Pressure means that time experienced by an individual is scarce in relation to the task demands at hand. In this article, we summarize findings and provide practitioner takeaways based on a systematic review of existing literature. We find that most empirical evidence supports reduced quality, increased productivity, and negative effects on individuals under time pressure. Time pressure is caused by company culture, poor effort estimates, and project management. The effects of time pressure can be explained by Challenge and Hindrance time pressure, the Yerkes-Dodson Law and The Job Demands-Resources model. Finally we conclude the article by giving practitioner takeaways related to minimizing the negative effects of time pressure.

Index Terms—Time Management, Cost Estimation, Synthesis of Affective Behavior.

I. INTRODUCTION

Time pressure has been a concern in software engineering from the start. In the 1970s, Frederick Brooks phrased what is now called *Brooks' Law*: “Adding manpower to a late software project makes it later.” Additionally, project overruns and overtime work have been reported to occur frequently in software development. As time pressure has been defined as the perception that time is scarce in relation to the demands of the task [1], it is reasonable to assume that time pressure commonly occurs in the software engineering field.

Given the prevalence of time pressure¹ in software engineering, our aim in this article is to provide practitioners with a summary of the topic and takeaways based on our systematic review. In the review, we searched 5,414 papers and found 102 articles related to time pressure in software engineering. Details of this 19-page review, which includes 148 references, can be found in the authors' 2020 publication [2].

Theory building in Software Engineering has been advocated and its usefulness argued for both industry and research by Sjøberg et al. [3]. The perceived benefit of theory building, specifically for the industry and practitioners, is that it provides input for decision-making in resource management, and improves understanding

¹In this paper, time pressure refers to both budget pressure, e.g. eight hour (budget) to do a task, and schedule pressure, e.g. a task that must be done by tomorrow.

and prediction in a given setting, in this case software engineering. We believe our article achieves both of these goals. Thus, the novelty of this work lies in the fact that it combines different theories of time pressure from various fields, and results in a unified overarching view of the topic within the context of the field of software development in Figure 1.

Figure 1 is used throughout this article to explain Causes (Section III), Theory (Section IV), and Effects (Section V) of time pressure in software engineering. Finally, after explaining the empirical findings on time pressure, we provide practical takeaways in Section VI.

II. METHODOLOGY

Because terminology surrounding the subject of time pressure varies widely throughout literature, our review protocol necessitated a complete reading of article abstracts for the majority of search results to identify relevant research. The search included Scopus and Google Scholar databases and snowballing—investigating articles citing and cited by a particular source.

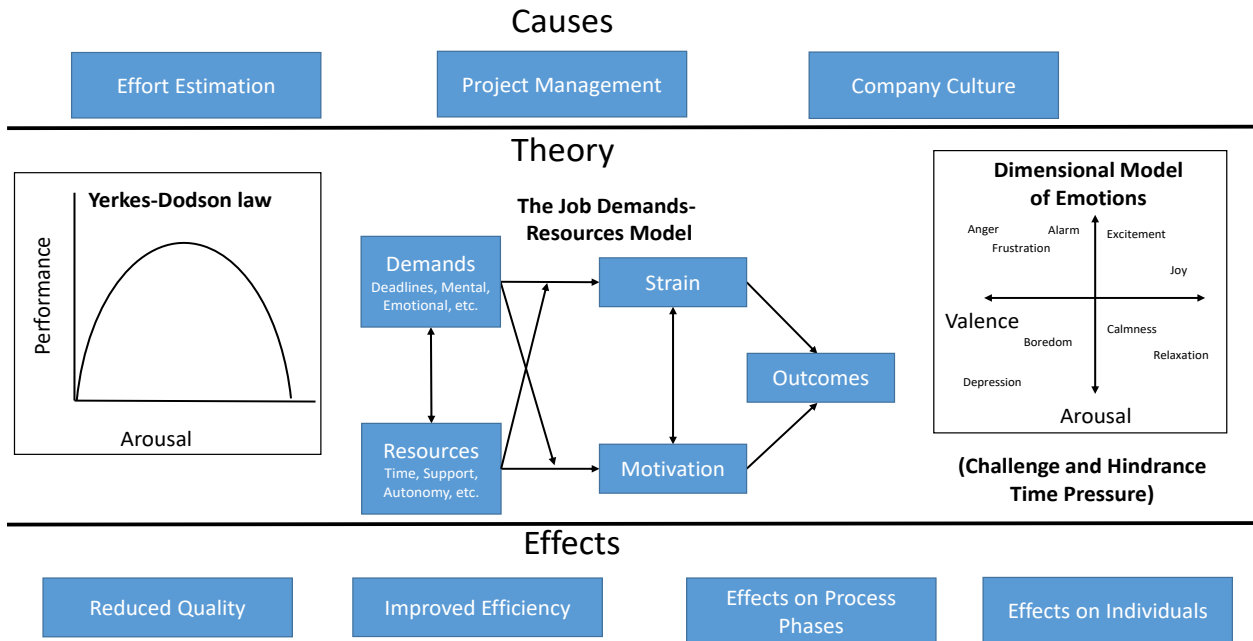
In our original article [2], our research questions focused on mapping existing definitions, studied process phases and approaches, metrics, and causes of time pressure in software engineering. Our goal is to present the causes, theory, and effects of time pressure in a software engineering setting by positioning our findings within a theoretical framework and by providing practical takeaways.

III. CAUSES

Causes of time pressure can be technical, social, or a combination thereof. Time pressure from technical causes arises as a result of errors in work *effort estimation* or *project management* and planning; social causes refer to situations where the time pressure is a consequence of leadership style and what is valued in the *company culture*.

First, problems in effort estimation can cause time pressure when insufficient time for the task is available [4]. Poor effort estimates can be caused by a lack of historical data, which can be the product of failure to collect data at all, or uniqueness in the scope or features

Fig. 1. Diagram describing causes, theory, and effects of time pressure in the context of software engineering.



of the project. Business motivations for earlier deadlines can affect effort estimation and cause time pressure. Such motivations may consist of contractual obligations to provide a solution before a due date or changes in laws such as the recent General Data Protection Regulation (GDPR) implemented by the European Union. Effort estimates can be improved with historical data and better models, whereas deadlines outside the control of the project's personnel can usually only be met with overtime work.

Second, poor project management can cause time pressure [5]. Task switching, poor organization, and a lack of buffer time in case of unanticipated work (e.g., change requests) or unforeseen resource shortages (e.g., unplanned leaves) cause time pressure, if they are not taken into account when managing the project. This kind of time pressure can be avoided by having proper risk management and slack time in plans, which can be later dedicated to unexpected work or resource shortages. In scenarios where task switching necessarily occurs, it also helps to let employees know which task to prioritize.

Third, company culture can cause time pressure [6]. Setting strict deadlines one after another without recovery time can lead to an unsustainable pace of development and a constant crisis mentality, where long-term planning and management are crushed under the pressure to meet the next deadline. This creates a company culture where long hours or constant deadline pressure are the

norm, rather than shortcomings of development process, effort estimation, or project management. Long working hours and meeting tight deadlines can be seen as a commitment to the company, a way to advance one's career, and in a sense, a performance measure in itself.

Prioritizing commitment to the company and overtime work may shut down career advancement of individuals with other priorities, such as those with families and children. In a company culture where working long hours is prioritized, time pressure can be worsened by requiring effort estimations by employees themselves. In such cases, the worker will be left wondering whether to give an accurate estimate or an estimate that pleases their superiors.

IV. THEORY

Figure 1 introduces three theories that are used to conceptualize and explain the effects of time pressure, namely **the Yerkes-Dodson law**, **Challenge and Hindrance Time Pressure** and **The Job Demands-Resources Model**. In Figure 1, Challenge and Hindrance Time Pressure is illustrated using a **Dimensional Model of Emotions** [7].

Increased time pressure heightens an individual's level of emotional activity, or arousal, in which he or she is reactive to stimuli and mentally "awake." **The Yerkes-Dodson Law** refers to an inverted U-shaped relationship between performance and arousal, where performance

increases with arousal, but only up to a certain point. After a maximum level of performance has been achieved, additional increases in arousal have a negative effect on performance. This is partly due to the negative effects of stress on cognitive performance [8].

The Job Demands-Resources Model is widely cited framework for understanding occupational stress, motivation, and outcomes [9]. It generalizes every job as having demands and resources. In relation to time pressure, the next deadline can be seen as a demand, while time, given tools, support, and autonomy can be seen as resources. In the model, resources are seen to increase job motivation and decrease job strain. Similarly, job demands increase strain. This means that job resources, such as adequate time, autonomy, appraisal of job performance, and support from a superior can buffer the negative effects of job demands on job strain. Furthermore, the model demonstrates how, when job demands are high, job resources increase motivation. Finally, performance and organizational outcomes (such as the timely delivery of projects and resulting profit) are negatively affected by job strain and enhanced by employee motivation.

A dimensional model of emotions describes emotions along two axes: Valence and Arousal. Valence refers to the positive-negative dimension, while Arousal refers to activity level and wakefulness. In our review, we discovered the **Challenge and Hindrance Time Pressure** framework proposed by D.S Chong et al . [10], in which challenge time pressure is viewed as encouraging and challenging—a situation of high Valence and high Arousal—situating it in the top right quadrant of the Dimensional Model of Emotions in Figure 1. Hindrance time pressure is viewed more negatively, and refers to situations of low Valence and high Arousal, such as the impossibility of completing the project in time or constant task switching under time pressure. As a result, both challenge and hindrance time pressure are high on the Arousal dimension, but are on the opposite sides of the valence axis. This framework has been used to explain some exceptional efficiency and performance under time pressure, such as the performance of the Apollo 13 ground crew.

To summarize, three theories that may be used to explain time pressure in the context of software engineering are as follows::

- Demands of the next deadline increase emotional arousal. **Yerkes-Dodson’s Law** focuses on how emotional arousal affects performance using an inverted U-shaped model;
- **Challenge and Hindrance Time Pressure** focuses on how arousal is perceived by the individual;
- **The Job Demands-Resources Model** highlights the motivating and buffering effects of job resources

on time pressure.

V. EFFECTS

Four effects of time pressure are shown at the bottom of Figure 1. We can see four types of effects: *Effects on Individuals*, *Effects on Process*, *Reduced Quality*, and *Improved Efficiency*.

a) Effects on Individuals: Multiple sources identified in our study link extreme and prolonged time pressure to several negative effects on individual well-being in the software development context. Time pressure has been shown to decrease confidence, and increase feelings of uneasiness and the willingness to both postpone decisions and re-use code unethically. Time pressure is also reported to be the most frequent external cause of unhappiness among developers. Most importantly, time pressure has been linked to burnouts and depressive symptoms, which have in turn been linked to clinical depression.

It has been reported that an individual’s knowledge plays a role in the mediation of time pressure, where an expert’s performance either increases or does not suffer as much as that of a novice. Similarly, a study has highlighted that projects with schedule pressure benefit from managers who score high in practical intelligence, which is defined as the ability to resolve “unexpected and difficult situations that often cannot be solved using established processes and frameworks” [11]. Benefits mentioned in the study were lower cost and higher customer satisfaction compared to other projects.

b) Effects on Process: In our review [2], we mapped all found sources to process phases, processes, or approaches used in the field of Software Engineering. We present the found categories in Table I. The most common sub-category was “Cost estimation, cost models, simulation and project escalation,” which includes multiple sources developing and evaluation of software project cost models and their schedule compression trade-offs. Consequently, these papers are related to a major cause of time pressure: errors related to cost estimation and scheduling.

The process phase where the impact of time pressure was the most often reported, measured with the number of papers, was “Quality Assurance.” We suspect that this is in part because, in the traditional waterfall model of software development, testing is the last phase, where time pressure has both the effect of lower quality product being tested and less time for testing. The increased pressure might be because of schedule slips or a constrained original schedule. Additionally, under time pressure, short-term solutions are prioritized, and shortcuts are more easily taken. For example, several sources identified in our study reported that time pres-

TABLE I
FOUND SOURCES BY INVESTIGATED SOFTWARE DEVELOPMENT
PROCESS PHASES AND METHODOLOGIES

Category	Sub-Category	N
Process phase	Requirements Engineering	2
Process phase	Design and Acquisition	5
Process phase	Programming and Implementation	5
Process phase	Quality Assurance	16
Whole Process or approach	Evolution and Maintenance:	3
Whole Process or approach	Agile and Scrum:	9
Whole process or approach	Process Improvement	7
Whole process or approach	Cost Estimation, Cost Models, Simulation and Project Escalation:	29
Whole Process or Approach	Project Success and Failure	5
Other	Detection of Time Pressure	3
Other	Group Interaction	5
Other	Fields other than Software Engineering and Information Systems	13
Other	Literature Reviews and Theoretical Papers:	6
Other	New Product Development	3
Other	Individual Psychological Factors	7

sure was obstructing the adoption of software process improvement (SPI) practices.

c) Reduced Quality and Improved Efficiency up to a point: In our systematic review [2], we extracted evidence related to time pressure with respect to outcome in terms of quality and efficiency.

Lastly, row Efficiency - Both / U-shape means that time pressure, depending on the situation, either increases or decreases efficiency. The inverted U-shaped relationship is one typical example of this, which means that time pressure increases efficiency up to certain point but after that increases in time pressure cause decrease in efficiency, (see Yerkes-Dodson Law in Figure 1).

Seven empirical studies support an increase in development efficiency due to time pressure, two papers report an inverted U-shaped relationship according to the Yerkes-Dodson Law, and two papers report either a decrease or no effect on efficiency. One variable that may create a situation in which time pressure has no effect on efficiency is the nature of the task. For example, in a database development task, participants were simply unable to go faster even when under time pressure [12]. Thus according to gathered evidence, small to medium size time pressure increases efficiency in software devel-

TABLE II
EFFECT OF TIME PRESSURE

Assumption	Empirical studies	Models
Efficiency - Increase	7	2
Efficiency - Decrease	2	8
Efficiency - Both / U-shape	2	2
Quality - Increase	4	-
Quality - Decrease	9	6

opment. Some studies have also captured the negative efficiency as predicted by the Yerkes-Dodson Law, i.e., when exceeding optimal time pressure leads to decreased efficiency.

Nine empirical studies support a reduction in software quality due to time pressure, while four studies reporting an increase in quality. The four latter studies had either low statistical significance or achieved statistically significant correlation, but did not use more robust multiple regression, making the finding questionable. Cost estimation and scheduling models unanimously support a decrease in quality under time pressure (schedule compression).

Surprisingly, decreased efficiency under time pressure is assumed for eight different cost estimation and process simulation models, which is in conflict with empirical studies. Increase in efficiency is assumed by two models. Two models also report that both increased and decreased efficiency outcomes are possible: time pressure is modeled to increase work progress efficiency, but also to increase error rate, and thus a decreasing maintenance efficiency.

We suspect that the partly differing results of the studies in Table II, can be explained by several factors, as follows. First, the project duration, as well as the time scales used, can differ from project to project and from study to study. We suspect that shorter projects have more positive effects of time pressure. Second, the starting points of the projects can also differ, and if the staff is already exhausted from the time pressure of the previous project, then time pressure in the next project might have a negative effect. Also, how time pressure is addressed or perceived can have an impact, Challenge Hindrance time pressure model on the previous Section explains how time pressure can be viewed as positive or negative. Managerial reaction to time pressure may also differ. If time pressure is achieved by more overtime hours, then individual well-being will decrease while maintaining project efficiency. If time pressure is achieved by taking shortcuts, the efficiency will decrease as a result of rework.

VI. PRACTITIONER TAKEAWAYS

Time pressure is common in the software industry. It is caused by effort underestimation, poor project planning and management, and company culture, while commercial pressure undoubtedly impacts all of the causes. Scientific evidence strongly points toward increased efficiency and reduced quality of software under time pressure. On the positive side, time pressure can force a development team to focus on the paramount features of a product. On the negative side, time pressure can induce tunnel vision where making the next deadline is prioritized over product quality and worker well-being. Under time pressure, problems are fixed with quick makeshift patches that save effort in the short-term only. For example, implementing a single patch at least 12 times because no one had the time to make a proper single fix [13]. Hence, there exists an inherent trade-off, since under time pressure one can only prioritize either efficiency or quality.

In our review, we found evidence that the type of task plays a role in this efficiency–quality trade-off. Efficiency improved less and quality decreased more, in tasks with high algorithmic nature, for example, in the development of database queries [12]. Some sources have also reported a mediating effect of knowledge on time pressure, that is, that more experienced, and skillful workers are less affected by time pressure. Hence a skilled manager would avoid time pressure in work done by more novice developers and in more complex and precise work that is algorithmic in nature.

Company culture where long working hours are a sign of commitment, a way to advance one’s career, and where individual heroics just in time for a deadline are highly appreciated, is a recipe for chronic time pressure. While the advantages of time pressure in increased efficiency can impress decision-makers and managers at first, as per the Yerkes-Dodson Law, the benefits of efficiency start to decrease over time with prolonged time pressure and without recovery time. Additionally, the negative effects on well-being become apparent in the form of increased unhappiness, depressive symptoms, and burnouts. Finally, such culture is discriminatory to workers that have important commitments outside of work, for example those with small children.

In our view, the success of Agile software development is inherently linked to a small to medium amount of time pressure and constant deadline setting. Psychological experiments have shown that aggressive intermediate deadlines increase overall outcome quality and individual time management [14]. This is because they balance required efforts over the duration of the project, while a single deadline tends to result in a focus of effort towards the end of the deadline. Additionally, when testing and development phases overlap, the intense time

pressure related to testing a defect-ridden product close to release is avoided. Finally, according to the well-established occupational psychology theory of the Job Demands-Resources Model, high independence and decision latitude related to work, which have been reported in Agile teams, can reduce the effects of time pressure and other stressors. However, changing to Agile software development cannot solve all problems, as issues such as the lack of code refactoring and testing reported in this context [5].

REFERENCES

- [1] J. R. Kelly and J. E. McGrath, “Effects of time limits and task types on task performance and interaction of four-person groups,” *Journal of Personality and Social Psychology*, vol. 49, no. 2, p. 395, 1985.
- [2] M. Kuutila, M. Mäntylä, U. Farooq, and M. Claes, “Time pressure in software engineering: A systematic review,” *Information and Software Technology*, vol. 121, 2020.
- [3] D. I. Sjøberg, T. Dybå, B. C. Anda, and J. E. Hannay, “Building theories in software engineering,” in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 312–336.
- [4] C. Jones, “Social and technical reasons for software project failures,” *CrossTalk*, vol. 19, no. 6, pp. 4–9, 2006.
- [5] A. Deak, T. Stålhane, and G. Sindre, “Challenges and strategies for motivating software testing personnel,” *Information and Software Technology*, vol. 73, pp. 1–15, 2016.
- [6] L. A. Perlow, “The time famine: Toward a sociology of work time,” *Administrative science quarterly*, vol. 44, no. 1, pp. 57–81, 1999.
- [7] J. A. Russell, “A circumplex model of affect,” *Journal of Personality and Social Psychology*, vol. 39, no. 6, p. 1161, 1980.
- [8] S. J. Lupien, F. Maheu, M. Tu, A. Fiocco, and T. E. Schramek, “The effects of stress and stress hormones on human cognition: Implications for the field of brain and cognition,” *Brain and Cognition*, vol. 65, no. 3, pp. 209–237, 2007.
- [9] A. B. Bakker and E. Demerouti, “The job demands-resources model: State of the art,” *Journal of Managerial Psychology*, 2007.
- [10] D. S. Chong, W. Van Eerde, K. H. Chai, and C. G. Rutte, “A double-edged sword: The effects of challenge and hindrance time pressure on new product development teams,” *IEEE Transactions on Engineering Management*, vol. 58, no. 1, pp. 71–86, 2011.
- [11] N. Langer, S. A. Slaughter, and T. Mukhopadhyay, “Project managers’ practical intelligence and project performance in software offshore outsourcing: A field study,” *Information Systems Research*, vol. 25, no. 2, pp. 364–384, 2014.
- [12] H. Topi, J. S. Valacich, and J. A. Hoffer, “The effects of task complexity and time availability limitations on human performance in database query tasks,” *International Journal of Human-Computer Studies*, vol. 62, no. 3, pp. 349–379, 2005.
- [13] M. Lavallée and P. N. Robillard, “Why good developers write bad code: An observational case study of the impacts of organizational factors on software quality,” in *Proc. of the 37th Int. Conf. on Software Engineering*, vol. 1. IEEE Press, 2015, pp. 677–687.
- [14] D. Ariely and K. Wertenbroch, “Procrastination, deadlines, and performance: Self-control by precommitment,” *Psychological Science*, vol. 13, no. 3, pp. 219–224, 2002.