

Miikka Kuutila

TIME PRESSURE AND WELL-BEING IN SOFTWARE ENGINEERING

*EVIDENCE FROM SOFTWARE REPOSITORIES,
EXPERIENCE SAMPLING, AND PRIOR LITERATURE*

UNIVERSITY OF OULU GRADUATE SCHOOL;
UNIVERSITY OF OULU,
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

A

SCIENTIAE RERUM
NATURALIUM



ACTA UNIVERSITATIS OULUENSIS
A Scientiae Rerum Naturalium 765

MIIKKA KUUTILA

**TIME PRESSURE AND WELL-BEING
IN SOFTWARE ENGINEERING**

Evidence from software repositories, experience
sampling, and prior literature

Academic dissertation to be presented with the assent of
the Doctoral Training Committee of Information
Technology and Electrical Engineering of the University of
Oulu for public defence in Martti Ahtisaari auditorium
(L2), Linnanmaa, on 3 December 2021, at 12 noon

UNIVERSITY OF OULU, OULU 2021

Copyright © 2021
Acta Univ. Oul. A 765, 2021

Supervised by
Professor Mika Mäntylä
Doctor Maëlick Claes

Reviewed by
Affiliate Professor Andrew Begel
Professor Alexander Serebrenik

Opponent
Professor Robert Feldt

ISBN 978-952-62-3134-1 (Paperback)
ISBN 978-952-62-3135-8 (PDF)

ISSN 0355-3191 (Printed)
ISSN 1796-220X (Online)

Cover Design
Raimo Ahonen

PUNAMUSTA
TAMPERE 2021

Kuutila, Miikka, Time pressure and well-being in software engineering. Evidence from software repositories, experience sampling, and prior literature

University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering

Acta Univ. Oul. A 765, 2021

University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

Popular and academic sources have indicated that high-pressure work environments are commonplace in the software industry, leading to stress and burnout. One cause of stress is time pressure, not having enough time to complete a task at hand. In addition to effects on well-being, time pressure affects software development processes, productivity, and quality. Synthesising prior evidence and providing real-time data to managers could help to minimize the detrimental effects and optimize productivity.

This thesis aims to investigate and give a comprehensive view of the existing body of knowledge on the effects of time pressure in software engineering, including processes, methods, and individual developers. Additionally, we aim to investigate ways to link time pressure and work well-being to software repositories to understand the well-being of software developers better. The research consists of two branches: a review branch and a primary study branch. In the review branch, prior knowledge related to sentiment analysis and time pressure was analyzed with bibliometric studies, making a systematic map and a systematic literature review. Studies were conducted using software repository mining, sentiment analysis, experience sampling, and interviews in the primary study branch.

Results from the review branch indicate, among others, increased productivity and decreased quality under time pressure. The causes of time pressure can be divided into technical and social factors, with errors in cost estimation, project management, and company culture being the most common causes. The results from the primary study branch show the limiting effect of individual differences on the prediction of well-being. Other findings include the detection of work rhythms through mining time stamps of code commits and the prediction ability of chat activity over chat sentiment on developer productivity.

While the research for this thesis could not find clear links between repository variables and developer well-being that would work at a team level, possibilities to study these links further are established. Future work related to time pressure in software engineering should focus on contextual factors such as company culture and trade-offs between productivity, quality, and well-being within different time scales.

Keywords: field study, human factors, longitudinal study, repository mining, sentiment analysis, time pressure, work well-being

Kuutila, Miikka, Aikapaine ja hyvinvointi ohjelmistotuotannossa. Tutkimusaineistoa ohjelmistojen versionhallintatyökaluista, ESM-menetelmästä ja aikaisemmasta kirjallisuudesta

Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta

Acta Univ. Oul. A 765, 2021

Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä

Ammattilais- ja akateeminen kirjallisuus on viitannut painostavien työympäristöjen olevan yleisiä ohjelmistoalalla, johtaen ylimääräiseen stressiin ja työuupumukseen. Yksi stressin lähde on aikapaine, ts. tehtävän tekemiseen ei ole tarpeeksi aikaa. Heikentyneen työhyvinvoinnin lisäksi aikapaine vaikuttaa tuottavuuteen ja ohjelmistojen laatuun. Aikaisempien tutkimustulosten syntetisointi ja reaaliaikaisen tiedon tuottaminen managereille voisi helpottaa aikapaineen haitallisia vaikutuksia ja parantaa tehokkuutta.

Tämä väitöskirja yrittää antaa kokonaisvaltaisemman kuvan olemassa olevasta aikapaineeseen liittyvästä kirjallisuudesta ohjelmistokehityksen kontekstissa, mukaanlukien vaikutuksista prosesseihin, metodeihin ja ohjelmistokehittäjiin. Lisäksi tavoitteena on myös yrittää yhdistää aikapaine ja työhyvinvointi ohjelmistokehityksen työkaluista saatavaan tietoon. Tehty tutkimus koostuu kahdesta osiosta: kirjallisuuskatsaukset ja primääriset tutkimukset. Kirjallisuuskatsauksiin keskittyvässä osiossa käytettiin muunmuassa klusteriointia laajojen aineistojen katselmoimiseen liittyen sentimentti analyysiin ja aikapaineeseen. Lisäksi tehtiin systemaattinen kartta ja -katsaus aikapaineeseen ohjelmistokehityksen kontekstissa. Primääritutkimuksissa käytettiin tutkimusmetodologioina ohjelmistokehitykseen liittyvien tietolähteiden "louhintaa", sentimentti analyysiä, ESM-menetelmää ja haastatteluja.

Kirjallisuuskatsausosion tulokset näyttävät aikapaineen lisäävän tuottavuutta ja huonontavan laatua ohjelmistokehityksessä. Aikapaineen aiheuttajat ovat teknisiä ja sosiaalisia ja ne liittyvät kolmeen kategoriaan: virheet kustannusarvioissa, virheet projektijohtamisessa ja yrityksen kulttuuri. Primääritutkimusosion tulokset näyttävät, kuinka erot ohjelmistokehittäjien välillä vaikeuttavat hyvinvoinnin ennustamista ohjelmistokehitykseen liittyvistä työkaluista saadusta tiedosta. Muita tuloksia ovat se, että kommitoidun ohjelmakoodin määrä seuraa vuorokausirytmiiä avoimenlähdekoodin projekteissa, sekä se, että yksittäisessä ohjelmistoprojektissa kommunikaation määrä ennusti kommitoidun lähdekoodin määrää paremmin kuin kommunikaatiossa oleva sentimentti.

Vaikka tämä väitöstutkimus ei pystynyt löytämään ohjelmistokehitystyökaluista saatavien muuttujien ja ohjelmistokehittäjien hyvinvoinnin välille selviä linkkejä, jotka toimoisivat hyvinä ennustajina ohjelmistokehitystiimin tasolla, osoittaa tutkimus lisää mahdollisuuksia tutkia näitä linkkejä. Tulevan aikapaineeseen liittyvän tutkimuksen ohjelmistokehityksen saralla tulisi keskittyä kontekstisidonnaisiin muuttujiin, kuten yrityskulttuuriin, sekä valintoihin tuottavuuden, laadun ja hyvinvoinnin välillä eri aikajänteillä.

Asiasanat: aikapaine, kirjallisuuskatsaus, pitkäaikaistutkimus, sentimentti analyysi, työhyvinvointi

Acknowledgements

My Ph.D. journey has been a rewarding journey of personal growth and, sometimes, a character-building exercise. Conducting research has been an inherently motivating task, which has produced moments of flow state and pure joy and experiences of extreme time pressure and even despair. Going near my limits in my abilities has given me perspective, resolve, and self-knowledge, which will help me face challenges for the rest of my life. While overcoming difficult challenges and self-doubt has given me more self-confidence and self-reliance, this thesis would not be possible without plenty of support. Hence, my sincere gratitude is in order.

First, I am very grateful to my supervisor Professor Mika Mäntylä, who offered me this position in what feels like an eon ago. Without your guidance and support, this dissertation would not have been possible. I would also like to thank my co-supervisor, Dr. Maëlick Claes, for your steady ideas and invaluable help during my doctoral studies. I want to thank Professor Markku Oivo for leading a first-class research unit, where I felt welcome as a doctoral student. Thanks to my follow-up group members are also in order. Thank you, Professor Jouni Markkula and Professor Pilar Rodriguez, for asking the right questions about my research and future.

I also want to express my gratitude towards Professor Andrew Begel (University of Washington) and Professor Alexander Serebrenik (Eindhoven University of Technology) for examining my doctoral thesis. The interest and the small exchanges during the SEmotion workshop gave me positive feedback at the beginning of this journey. I am also very grateful to Professor Robert Feldt (Chalmers University of Technology) for his willingness to be an opponent during my doctoral defense.

I would also like to thank the co-authors for their expertise and input for the publications included in this thesis. In addition, I would like to thank Professor Bram Adams for hosting me for a research visit in Montreal in 2018. The visit was an unforgettable and tremendous experience for me. I would also like to thank Professor Marko Elovainio and Dr. Umar Farooq for their indispensable input regarding my research. I would also like to thank Dr. Daniel Graziotin for being a great person to work with. I would also like to thank Professor Filippo Lanubile and Professor Nicole Novielli for co-authoring the arousal lexicon, which was used to conduct some research for this thesis.

I would also like to thank Suomen Akatemia for funding my doctoral research, and KAUTE-säätiö for supporting my research visit to the Polytechnique Montréal, Quebec, Canada.

I would also like to thank all my colleagues at the M3S research group. I appreciate the peer support during Oulanka and Liminka seminars and regular coffee breaks, which increased my morale. I want to thank (in no particular order) Leevi, Murali, Elina, Prabhat, Yuqing, Shayan, Itir, Iflaah, Rahul, Alireza, Adrian, Woubshet, Päivi, Vitor, Nirnaya, Heidi, Ari, Juha, Anna, and both Pertti's. I would also like to thank all of my other colleagues and senior members of the M3S research group. Lastly, I want to thank my mother, Elina Romppainen, and my brother Janne Kuutila for their support.

Miikka Kuutila Oulu, Finland, October 2021.

List of original publications

This thesis is based on the following publications, which are referred to in the text by their Roman numerals (I–IX).

- I Kuuttila, Miikka., Mäntylä, Mika, Claes, Maëlick and Elovainio, Marko (2017). Reviewing literature on time pressure in software engineering and related professions: computer assisted interdisciplinary literature review. *IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. pp.54-59. IEEE. DOI: 10.1109/SEmotion.2017.11
- II Mäntylä, Mika., Graziotin, Daniel and Kuuttila, Miikka.(2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Computer Science Review.*, 27, pp. 16-32. DOI: 10.1016/j.cosrev.2017.10.002
- III Kuuttila, Miikka., Mäntylä, Mika., Farooq, Umar and Claes, Maëlick. (2020). Time Pressure in Software Engineering: A Systematic Review. *Information and Software Technology.*, 121. Elsevier. DOI: 10.1016/j.infsof.2020.106257
- IV Kuuttila, Miikka., Mäntylä, Mika, Farooq, Umar and Claes, Maëlick. (2020). What Do We Know about Time Pressure in Software Development? *IEEE Software.* 38 (5). pp.32-38. IEEE. DOI: 10.1109/MS.2020.3020784
- V Claes, Maëlick., Mäntylä, Mika., Kuuttila, Miikka and Adams, Bram. (2018). Do programmers work at night or during the weekend?. In *Proceedings of the 40th International Conference on Software Engineering*, pp. 705-715. DOI: 10.1145/3180155.3180193
- VI Kuuttila, Miikka., Mäntylä, Mika. and Claes, Maëlick. (2020). Chat activity is a better predictor than chat sentiment on software developers productivity. *IEEE/ACM 5th International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. pp.39-43. DOI: 10.1145/3387940.3392224
- VII Kuuttila, Miikka., Mäntylä, Mika., Claes, Maëlick and Elovainio, Marko. (2018). Daily questionnaire to assess self-reported well-being during a software development project. *IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. pp.39-43. DOI: 10.1145/3194932.3194942
- VIII Kuuttila, Miikka., Mäntylä, Mika, Claes, Maëlick., Elovainio, Marko and Adams, Bram. (2018). Using experience sampling to link software repositories with emotions and work well-being. *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. pp.29. ACM. DOI: 10.1145/3239235.3239245
- IX Kuuttila, Miikka., Mäntylä, Mika, Claes, Maëlick., Elovainio, Marko and Adams, Bram. (2021). Individual Differences Limit Predicting Well-being and Productivity Using Software Repositories: A Longitudinal Industrial Study. *Empirical Software Engineering*, 26(5), pp.1-30. Springer. DOI: 10.1007/s10664-021-09977-1

Contents

Abstract	
Tiivistelmä	
Acknowledgements	7
List of original publications	9
Contents	11
1 Introduction	15
1.1 Motivation and research gaps	16
1.2 Research questions	17
1.3 Dissertation structure	17
2 Background	19
2.1 Time pressure	19
2.2 Work well-being	22
2.3 Repository mining	23
2.4 Sentiment analysis	24
2.5 Affective computing	25
2.5.1 Measurements	26
2.5.2 Interventions	29
3 Methodology	31
3.1 Research process	31
3.2 Reviews	31
3.2.1 Bibliometric studies complemented with natural language processing — Papers I and II	31
3.2.2 Systematic maps and reviews — Papers III and IV	34
3.3 Primary study branch	35
3.3.1 Repository mining — Papers V and VI	36
3.3.2 Field studies — Papers VII, VIII and IX	38
4 Original research papers	43
4.1 RQ1: What is the current state of research related to sentiment analysis and time pressure in the current literature?	43
4.1.1 Introduction	43
4.1.2 Paper I: Reviewing literature on time pressure in software engineering and related professions: computer assisted interdisciplinary literature review	44

4.1.3	Paper II: The evolution of sentiment analysis — A review of research topics, venues, and top cited papers.	46
4.1.4	Main contributions and findings.	46
4.2	RQ2: What is the current state of research related to time pressure in software engineering?	47
4.2.1	Introduction.	47
4.2.2	Paper III: Time pressure in software engineering: a systematic review.	48
4.2.3	Paper IV: What do we know about time pressure in software development?	51
4.2.4	Main contributions and findings.	52
4.3	RQ3: Can sentiment analysis and commit activity times explain developer productivity and behaviour?	52
4.3.1	Introduction	52
4.3.2	Paper V: Do programmers work at night or during the weekend? . . .	53
4.3.3	Paper VI: Chat activity is a better predictor than chat sentiment on software developers productivity.	55
4.3.4	Main contributions and findings.	57
4.4	RQ4: How does developer well-being link to software repository variables?	57
4.4.1	Introduction.	57
4.4.2	Paper VII: Daily questionnaire to assess self-reported well-being during a software development project.	58
4.4.3	Paper VIII: Using experience sampling to link software repositories with emotions and work well-being.	59
4.4.4	Paper IX: Individual differences limit predicting well-being and productivity using software repositories: a longitudinal industrial study	61
4.4.5	Main contributions and findings.	68
5	Discussion	69
5.1	RQ1 - What is the current state of research related to sentiment analysis and time pressure in the current literature?	69
5.2	RQ2 - What is the current state of research related to time pressure in software engineering?	70
5.3	RQ3 - Can sentiment analysis and commit activity times explain developer productivity and behaviour?	71

5.4	RQ4 - How does developer well-being link to software repository variables?	72
5.5	Threats to validity	74
5.5.1	Construct validity	74
5.5.2	Internal validity	75
5.5.3	External validity	76
5.5.4	Conclusion validity	76
6	Conclusions	77
6.1	Contributions	77
6.1.1	Future research	80
	References	83
	Original publications	97

1 Introduction

Modern software development is a form of demanding collaborative knowledge work. However, many articles in popular sources, such as Gaudin (2015) and Schreier (2016), tell a tale of a software industry riddled with pressured work environments. Indeed, even experiences of burnout from stress seem to be common according to industry sources such as Bradford (2018).

Concerning these human-related issues, there has been an increasing interest in the affect of software developers, and calls for studies in the field of *behavioural software engineering* by Lenberg, Feldt, and Wallgren (2015), as well as in the field of *psychoempirical software engineering* by Graziotin, Wang, and Abrahamsson (2015).

Interest in scheduling and time-related issues in software engineering has been expressed almost since the beginning. In the 1970s, in the widely influential *The Mythical Man-Month: Essays on Software Engineering*, Frederick Brooks coined the idea known as Brooks' law: "adding manpower to a late software project makes it later" (Brooks Jr., 1995). Similarly, textbooks from the '80s and '90s for software developers and managers have dedicated chapters and subchapters for "deadline pressure" by Gilb and Finzi (1988) and "beating schedule pressure" by McConnell (1996). More recently, it has been shown that 60-80% of software projects are late (encounter overruns) (Molokken & Jorgensen, 2003). Because being late is one antecedent of time pressure, we can assume the latter is reasonably common in the software industry.

In psychological literature, time pressure refers to situations where time is seen as a limited and scarce resource (Maule, Hockey, & Bdzola, 2000). Time pressure has thus been defined as the perception that time is scarce in relation to the demands of the task (e.g., Basten (2017), Kelly and McGrath (1985), and Cooper, Dewe, and O'Driscoll (2001)).

Subjective well-being has been described as a broad category of phenomena, including people's emotional responses, domain satisfactions, and global judgments of satisfaction (Diener, Suh, Lucas, & Smith, 1999). Moreover, Diener et al. (1999) define subjective well-being as a general area of scientific interest; hence, each specific construct related to it needs to be understood individually. One of these constructs is work well-being. It is discussed at length by Schulte and Vainio (2010), who point to a positive relationship between work well-being and productivity at a societal level. In occupational psychology, the well-known job demands-resources model is used to explain employee well-being (Karasek and Theorell (1990) and Bakker and Demerouti (2007)). Generally, the model assumes that every job has both demands and resources,

and proper balancing of them leads to the well-being of the employee. Resources in the model can refer to skills, autonomy, feedback, and others, while demands can include role ambiguity, performance, and emotional demands. Hence, from this point of view, time pressure can be seen as a demand. Hence, too much time pressure leads to worse well-being outcomes, stress, exhaustion, and even burnout (Demerouti, Bakker, Nachreiner, and Schaufeli (2001) and Svenson (1993)).

Recently, software repository mining has become a popular way for acquiring data for software engineering research (de F. Farias et al., 2016). According to Hassan (2008), these repositories include source control repositories, bug repositories, archived communications, deployment logs, and code repositories. Studies on mining software repositories have made several recent attempts to build tools and reason about the affective states of software developers by utilizing sentiment analysis (*e.g.*, Mäntylä, Novielli, Lanubile, Claes, and Kuutila (2017) and Novielli, Girardi, and Lanubile (2018)).

Sentiment analysis has been defined as a series of methods, techniques, and tools for detecting and extracting subjective information from language, such as opinions and attitudes (Liu, 2009). In the software engineering context, Jongeling, Datta, and Serebrenik (2015) have compared and evaluated general sentiment analysis tools and their performance. The results were that the tools did not agree with each other or with human manual labeling, thus concluding that tools for software development specific context are needed.

1.1 Motivation and research gaps

While plenty of academic literature is partially or wholly related to the causes and effects of time pressure, attempts to summarize this knowledge systemically and in a comprehensive way have been limited. Thus, efforts for bibliometric studies, a systematic mapping and a systematic review of the area are justified.

Software repositories contain a multitude of data about software developers. This includes the amount of work measured in number of commits and lines of code changed, working times in time stamps and sentiment in the form of natural language text. While a multitude of studies have looked at these topics individually, there has not been an effort to link software repository variables to software developer's well-being, and thus to the detection of time pressure in the software development context. Detection of time pressure from software repository variables would allow for real-time monitoring of project status, and in the best case improvements on both development efficiency and developer well-being.

1.2 Research questions

Based on the identified research gaps in the preceding subsection, the following research questions are formed:

- RQ1: What is the current state of research related to sentiment analysis and time pressure in the current literature? (Papers I and II)
- RQ2: What is the current state of research related to time pressure in software engineering? (Papers III and IV)
- RQ3: Can sentiment analysis and commit activity times explain developer productivity and behaviour? (Papers V and VI)
- RQ4: How does developer well-being link to software repository variables? (Papers VII, VIII, and IX)

1.3 Dissertation structure

The structure of this dissertation is as follows. Chapter 2 discusses the background in relation to the introduced research questions. Next, Chapter 3 provides details of the research process and the research methods used. Chapter 4 summarizes the motivations and results of the original studies. Chapter 5 discusses the findings and the validity threats to these findings, and answers the research questions. Lastly, Chapter 6 lists the contributions of the studies and discusses future directions and avenues for research.

2 Background

In this chapter, key concepts, background and related work relevant to the research questions shown in Chapter 1 are further developed. Section 2.1 introduces theory, metrics, and operationalizations related to time pressure. Section 2.2 presents theory related to work well-being. Next, sections 2.3 and 2.4 overview literature related to sentiment analysis and repository mining, respectively.

2.1 Time pressure

Time pressure refers to situations where time is seen as a limited and scarce resource related to task demands (Kelly & McGrath, 1985). The effects of scarcity of resources on human behavior is an active research area, as whole books have been written about the effects of scarcity mindset on humans (Mullainathan & Shafir, 2013). This scarcity mindset leads to tunneling and focus mindsets, where short-term goals are prioritized over long-term goals (Mullainathan & Shafir, 2013; Shah, Mullainathan, & Shafir, 2012). Thus, specifying these results to the software engineering context, scarcity of time could lead to focusing on quickly implemented solutions with technical debt over solutions guarding the longevity of the developed software. On the other hand, this could decrease gold plating, that is, working on an implementation or module more than is expected or needed.

Various definitions for emotional arousal exist, though activation of the nervous system is a common one (Storbeck & Clore, 2008). Yerkes Dodson's law (Yerkes & Dodson, 1908) is a relationship between arousal and performance found by psychologists Robert M. Yerkes and John Dillingham Dodson in 1908. It dictates that because of arousal, performance increases, but when arousal grows too much or lasts too long, performance starts to decline. On the other hand, without arousal, performance also declines. This relationship between performance and pressure dictated by Yerkes Dodson's law is presented in Figure 1.

The speed-accuracy trade-off Heitz (2014) is a ubiquitously observed phenomenon, meaning it has been observed with species such as ants or bumblebees. It offers another way to view time pressure: it observes that the decision speed is negatively correlated with decision quality. The speed-accuracy trade-off is used as a benchmark for decision processes across task domains, when examining only reaction speed or quality alone is not sufficient.

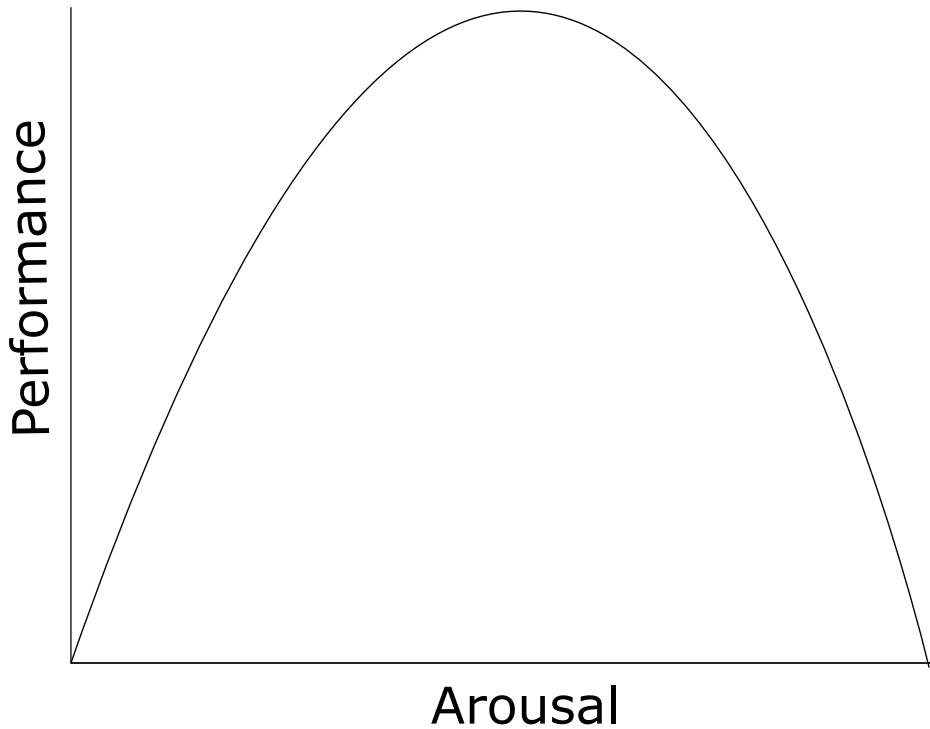


Fig. 1. The Yerkes-Dodson law. (Reprinted, with permission, from Paper III. © 2020 Elsevier).

Evidence for exceptional performance under extreme time pressure exists, such as the Apollo 13 ground crew (Chong, Van Eerde, Chai, & Rutte, 2011). Partly based on this observation, the challenge-hindrane framework (LePine, Podsakoff, & LePine, 2005) has been introduced. The framework assumes time pressure as having either positive (challenge) or negative (hindrance) effects on goal achievement. LePine et al. (2005) pointed out that challenges could be viewed as good stress, while hindrance could be viewed as bad stress (*i.e.*, distress). Thus, not only the amount, but also the type of time pressure matters. Examples from Chong et al. (2011) of hindrance (bad) time pressure are “amount of constant switching between tasks” or “impossibility to fulfill the project schedule,” while examples of challenge (good) time pressure are “importance of completing this project on time” or “urgent need for successful completion of the work the team is doing.” Chong et al. (2011) recognized that the boundaries between challenge and hindrance are not always clear, and there were several items that did not clearly fall in either category in their survey. In software engineering, the challenge-hindrane time pressure definition has been used by Lohan, Acton, and Conboy (2013) to study the

Table 1. Metrics identified from literature related to software development. (Modified slightly from Paper III. © 2020 Elsevier).

Metric to Measure Time Pressure	Example Paper	Data Source
Estimated time - customer time, divided by estimated time	Nan and Harter (2009)	Company Project Database
Actual schedule divided by nominal schedule	B. Boehm et al. (2000)	Company Project Database
Estimated effort minus remaining effort, divided by remaining effort	Ruiz et al. (2001)	Model simplification
Standard deviation of tasks completed in a project each month	Cataldo (2010)	Company Project Databases
Questionnaires and surveys	Maruping et al. (2015)	Questionnaires and surveys
Physiological measurements	Tuomivaara et al. (2017)	Skin Conductance Electromyography Electrocardiography, etc.
Sentiment analysis	Mäntylä et al. (2017)	Natural Language Text

effect of time pressure on group decision quality, showing both positive and negative effects.

Numerous terms exist that refer to situations with time pressure and are used synonymously with time pressure. These include deadline pressure (Ebert & Jones, 2009), schedule pressure (Costello, 1984) and time budget pressure (Nan & Harter, 2009). Another related term is schedule compression, which B. W. Boehm (1981) defined as the percentage of schedule cut in a project's planned duration compared to the nominal schedule of the project. Research paper III belonging to this thesis (Kuutila, Mäntylä, Farooq, & Claes, 2020) collects metrics used to measure time pressure in Table 1. These include different ratios related to time, schedules and effort, as well as using completed tasks, questionnaires, surveys, physiological measurements and sentiment analysis to measure time pressure.

Experimental designs used to create time pressure are shown in Table 2. They represent essential information on the operationalizations of time pressure in experimental settings, which was already noted in the nineties by Hwang (1994). In addition to using time limits, different task difficulties and improved rewards for faster completion have also been used in the literature.

Table 2. Creating time pressure in Software Development related Experiments. (Modified slightly from Paper III. © 2020 Elsevier).

Creating Time Pressure	Example Papers	Data Source
Time limits in experimental settings	Juristo and Vegas (2011)	performance in the experiment
Task difficulties in experimental settings	Ramanujan et al. (2000)	performance in the experiment
Reward for faster completion in experimental setting	Mäntylä et al. (2014)	performance in the experiment

A number of studies have focused on the relation between time pressure and software development productivity and quality. Again, Paper III (Kuutila et al., 2020) reviewed these effects in a systematic manner. The majority of the empirical studies reviewed show an improvement in efficiency but also lowered quality. This was partly contradicted by software project simulation models, the majority of which assume schedule compression increasing the total needed effort. Our review also found multiple studies linking time pressure to well-being, more of these in the next subsection.

Studies have also looked into specific software development tasks under time pressure. Recent studies by Salman, Turhan, and Vegas (2019) and Salman, Rodriguez, Turhan, Tosun, and Gureller (2020) have looked into time pressure in relation to confirmation bias in software testing, concluding that time pressure is an antecedent to confirmatory behaviour. This matches well with the scarcity-induced tunneling mindset discussed by Mullainathan and Shafir (2013).

2.2 Work well-being

The well recognized job demands-resources model (Bakker & Demerouti, 2007) is used to explain employee well-being in occupational psychology. The model assumes well-being to be the result of the balance between demands and resources. Resources in the model refer to skills, autonomy, feedback, and others, while job demands include role ambiguity, performance, and emotional demands. This in turn ties the model into time pressure, as it can be seen as a demand. Hence according to the model, too much demands including time pressure leads to worse well-being outcomes, exhaustion, and even burnout (Demerouti et al., 2001).

Subjective well-being was defined by Diener et al. (1999) as a broad category of phenomena, which includes people’s emotional responses, domain satisfactions, and global judgments of satisfaction. Furthermore, Diener et al. (1999) describe subjective well-being as a general area of scientific interest, where each specific construct related

to it needs to be understood individually. Work well-being is one such construct, which is discussed at length by Schulte and Vainio (2010). They describe work well-being as a summative concept that takes into account work satisfaction and occupational safety and health aspects. Schulte and Vainio (2010) also speculate that work well-being is linked to productivity at the individual, enterprise and societal levels.

Sonnentag, Brodbeck, Heinbokel, and Stolte (1994) surveyed 180 software developers to identify factors related to burnouts, discovering a lack of identification, (*i.e.*, praise and recognition, and perceived pressures such as time pressure) to be related to stressors. Similar results have been obtained by Singh and Suar (2013), who surveyed Indian software developers and found mediating effects to stress with subjective well-being, social support, and meditation. Fucci, Scanniello, Romano, and Juristo (2018) investigated the effect of sleep deprivation on software developers and found that a single night of sleep deprivation already had a negative effect on software quality.

2.3 Repository mining

According to Hassan (2008), repositories that can be mined for data include source control repositories, bug repositories, archived communications, deployment logs, and code repositories. Mining Software Repositories is a premier conference for publishing studies using repository mining in a software context (MSR, 2021).

A taxonomy for categorizing mining software repositories was presented by Kagdi, Collard, and Maletic (2007), specifically in the context of software evolution. This taxonomy takes into account what, why, how, and quality attributes related to MSR studies. Thus, the taxonomy lists the type of software repositories mined (what), the purpose (why), the methodology used (how), and the evaluation method of results.

According to the systematic mapping study by de F. Farias et al. (2016), the most popular data sources have been code, commit data and bug reports. Additionally, the most common interests of these studies have been software defects and software developer behaviour, while the broader research methodology has been either a case study or exploratory study.

Developing tools for mining this information has also been an active research area, and one such tool is Pydriller developed by Spadini, Aniche, and Bacchelli (2018). Data in the form of natural language text acquired with repository mining can be analyzed with methods such as sentiment analysis (*e.g.*, Romano, Caulo, Scanniello, Baldassarre, and Caivano (2020)).

2.4 Sentiment analysis

The Merriam-Webster dictionary (Merriam-Webster-Dictionary, n.d.) has multiple definitions for sentiment, including: predilection, opinion, emotion, or an idea colored by emotion. Sentiment analysis has been defined as a series of methods, techniques, and tools for detecting and extracting subjective information, such as opinions and attitudes, from language (Liu, 2009). According to Paper II (Mäntylä, Graziotin, & Kuutila, 2018), its breakthrough came in the mid-2000's by analyzing and synthesizing the results of movie reviews, and its growth from 2004 to 2016 was exponential.

Measuring and analyzing sentiment is thus closely related to theories of emotion. A multitude of theories of emotion exist, but they can be divided into discrete and dimensional models of emotion (Barrett, 1998). In discrete or categorical models of emotion, there are a limited number of basic emotions distinct from each other, and each has its own properties that distinguish it from other emotions. Seminal work by Paul Ekman has argued for basic emotions (Ekman, 1999). On the other hand, dimensional models of emotion place emotions in one or more dimensions on a scale. An example of such a circumplex model of emotions by Russell (1980) places all emotions on two dimensions: valence and arousal. Valence refers to placing emotions on an axis from negative to positive, while arousal refers to placing emotion on a low to high activity level.

Core affect, emotions, moods, and well-being are distinct but related concepts, and are introduced in Table 3. Russell and Barrett (1999) define core affect as a “neurophysiological state consciously accessible as a simple primitive non-reflective feeling most evident in mood and emotion but always available to consciousness”, while an occurrence of an emotion is “a complex set of interrelated sub-events concerned with a specific object”. Ekkekakis (2012) sees one defining aspect of moods is that they last longer than emotions. Moods are also seen to be global and diffused rather than specific (Ekkekakis, 2012). As Diener et al. (1999) define subjective well-being as “people’s longer-term affect, the lack of unpleasant affect and life satisfaction,” it can be said that positive and negative emotions are components of subjective well-being. Some debate exists over the duration of these concepts. Cultivating positive emotions is also advocated to be used in therapy by Fredrickson (2000) as an intervention strategy to “problems rooted in negative emotions, such as anxiety, depression, aggression and stress relate health problems”.

As emotions are part of subjective well-being, detecting sentiment could improve well-being. Furthermore, mood at the time of judging one’s well-being does have an effect on the judgment of one’s subjective well-being Schwarz and Clore (1983).

Table 3. Definitions and examples of distinguishing aspects of concepts related to emotional theories.

Concept	Duration or time scale	One of distinguishing aspects	Source paper
Core Affect	Any particular point in time.	Can occur in pure isolated form.	Schutz et al. (2007), Russell (2003)
Emotion	From minutes to hours.	Directed at an object.	Verduyn et al. (2015)
Mood	Longer than emotions, typically hours to days.	Not directed at an object. Prolonged core affect.	Ekkekakis (2012), Russell (2003)
Well-being	Longer term affect, typically weeks to months.		Diener et al. (1999), Schwarz and Clore (1983)

In the software engineering context, studies such as Jongeling et al. (2015) and Lin et al. (2018) have compared and evaluated general sentiment analysis tools and their performance specifically in the software engineering context, discovering that the tools evaluated did not agree with each other or manual labeling, thus concluding that tools for software development specific context are needed. More tools have indeed been implemented, such as Senti4SD by Calefato, Lanubile, Maiorano, and Novielli (2018). Software engineering and platform-specific fine tuning of tools has been recommended based on the comparisons of sentiment analysis tools by Novielli, Calefato, Lanubile, and Serebrenik (2021). Lexicon-based approaches have also been advocated over machine learning techniques in the absence of labeled data (Novielli, Calefato, Dongiovanni, Girardi, & Lanubile, 2020).

2.5 Affective computing

The field of affective computing has been defined as "computing that relates to, arises from, or deliberately influences emotions" by Picard (1997), or "trying to assign computers the human-like capabilities of observation, interpretation and generation of affect features." by Tao and Tan (2005). Thus, to observe and detect affective states, various measures and metrics are needed in the application of affective computing. This subsection first introduces commonly used measures in affective computing, and in the second part briefly introduces attempts to use interventions to mediate the effects of stress.

2.5.1 Measurements

An early seminal work to collect data with physiological sensors relating to affective states was done by Picard, Vyzas, and Healey (2001). In relation to affective computing, a review on the detection of stress has been published by Greene, Thapliyal, and Caban-Holt (2016). They classify physiological measures of stress to technologies as follows: brain activity with electroencephalography (EEG), heart activity with electrocardiography (ECG), skin response with galvanic skin response (GSR) and electrodermal activity (EDA), blood activity with photoplethysmography (PPG), muscle activity with electromyography (EMG), respiratory response with piezoelectricity/electromagnetic generation. Classification of physical measures is correspondingly facial expressions with automated facial expression analysis, eye activity with infrared eye tracking, and body gesture with automated gesture analysis. Tables 4 and 5 collect these measures of affective states by Greene et al. (2016) and augment them with other technologies and measures related to work tasks and human behaviour, which have been used in affective computing research. For each metric, we also give example study and usage. However, the list cannot be taken as complete, as no systematic efforts have been made in order to collect it.

A common way to detect cognitive stress in the literature is heart rate variability, that is, measuring the variance in the intervals between heartbeats. Seminal paper by McDuff, Gontarek, and Picard (2014) has demonstrated that heart rate variability measurements are also possible with a camera.

Skin conductance is used by Hernandez, Morris, and Picard (2011) together with self-reports to detect stress for call center employees. Individual differences show, as the individual models show an accuracy around 78%, which falls to 73% when using data from all participants. Multiple physiological sensors, including electrocardiography, are used by Healey and Picard (2005) for detecting stress during real-world driving tasks. Analysis includes distinguishing three levels of stress with an accuracy of 97%. Healey and Picard (2005) also correlate physiological data with a continuous metric of observable stressors, showing that in their study skin conductivity and heart rate through electrocardiography perform the best in detecting stress over respiration and electrical activity in skeletal muscles through EMG. The study by Chittaro and Sioni (2014) used EMG for detecting stress with a placebo, a single and multiple sensors setting during a relaxation game. The results demonstrate the importance of more thorough strategies in evaluating the usefulness of computing-based interventions.

Cho, Bianchi-Berthouze, and Julier (2017) developed a deep learning model recognizing stress levels from breathing patterns. Using a thermal camera, the deep

learning model is able to achieve 84.59% accuracy in detecting two levels of stress and an accuracy of 56.52% for three levels of stress.

Some less common metrics are electroencephalography and photoplethysmography, which measure electrical activity on the scalp and blood volume change over time, respectively. Bozhkov, Georgieva, Santos, Pereira, and Silva (2015) built a valence recognition system using electroencephalography (EEG) data. The main problem in using EEG data is building subject-independent models, which Bozhkov et al. (2015) solve with sequential feature selection (SFS) in their models. A very recent study by Perpetuini et al. (2021) demonstrated the viability of PPG in affective computing, as it was able to predict state anxiety with some accuracy. Task-evoked pupillary response was used by Iqbal, Zheng, and Bailey (2004) to detect mental workload during a task, with results showing more difficult tasks having a longer processing time, higher subjective rating of mental workload, and a greater pupillary response in important subtasks.

Measures found in the literature that are more indirect and related to human behaviour include keystroke pressure and email usage. Mark et al. (2016) investigated email usage, concluding that the duration of work was associated with increased stress, while the types of interruptions to work and batching email work did not have an effect on stress. Hernandez, Paredes, Roseway, and Czerwinski (2014) used a pressure-sensitive keyboard in an experiment, showing that more than 79% of participants used significantly increased typing pressure in stressful situations.

Majaranta and Bulling (2014) introduced literature related to eye tracking in human computer interaction. A survey of technical approaches is complimented in part by application areas, the four application areas are: explicit eye input for implementing gaze-based command and control of applications, attentive user interfaces where eye-gaze information is used to modify the user interface (e.g. increasing image resolution where the user is looking), gaze-based user modeling for understanding user behaviour, cognition and intent and passive eye monitoring, where eye-tracking is only recorded without any immediate reaction to eye-tracking. Similarly, for body gesture detection, Noroozi et al. (2018) list three application areas: systems that detect their user's emotions, actual or virtual conversational agents expected to act similarly to humans based on emotions, and systems that feel emotions themselves. Historically, automated recognition of emotion from facial images in the wild has been a challenge, though works such as Pons and Masip (2017) have been recently making progress with deep learning.

Knowing when to interrupt a computer user is an important research direction in affective computing, as this has been done with audio and video data sources in the

Table 4. Physiological measures used in affective computing. List of metrics by Greene et al. (2016) is augmented with heart-rate variability and pupillary response.

Metric	Based on	Example usage or finding	Example papers
Electroencephalography (EEG)	Electrical activity on the scalp.	Detection and classification of experienced emotions in valence axis.	Bozhkov et al. (2015)
Electrocardiography (ECG)	Electrical activity (voltage) of the heart over time.	Detection of stress during real-world driving tasks	Healey and Picard (2005)
Photoplethysmography (PPG)	Blood volume change over time.	Estimation of state anxiety from PPG features of brachial and radial arteries.	Perpetuini et al. (2021)
Electromyography (ECG or EMG)	Electrical activity in skeletal muscles	Detection of stress during relaxation game	Chittaro and Sioni (2014)
Respiration variability	Respiratory response - rate of breaths over time	Stress detection over cognitive tasks	Cho et al. (2017)
Heart rate variability (HRV)	Time interval between heartbeats.	Measurement of heart rate variability with a camera.	McDuff et al. (2014)
Skin conductance/response	Electrodermal activity of the skin.	Detection of stress for call center workers.	Hernandez et al. (2011)
Task-evoked pupillary response	Dilation and constriction of the pupil by the nervous system.	More difficult tasks induces higher subjective ratings of mental workload and a greater pupillary response	Iqbal et al. (2004)

Table 5. Physical and behavioural measures used in affective computing literature. List of metrics by Greene et al. (2016) is augmented with less common examples such as keystroke pressure and email use patterns.

Metric	Based on	Example usage or finding	Example papers
Automated facial expression analysis	Computational interpretation of facial expressions from static images or video sequences	Classification of facial expressions to discrete emotions	Pons and Masip (2017)
Eye tracking	Computational tracking of human's gaze	User's input in human computer interaction.	Majaranta and Bulling (2014)
Body gesture analysis	Computational interpretation of body gestures	User's emotion detection	Noroozi et al. (2018)
Keystroke pressure	Pressure applied to a keyboard while typing.	Detection of stress for computer users	Hernandez et al. (2014)
Email use patterns	Patterns of duration, interruptions and batching of emails during work	Duration of time spent on email work associated with increased stress	Mark et al. (2016)

seminal work of James Fogarty and Scott Hudson. Features and events such as whether any occupant has talked, is positioned at their desk, whether guests are present, which time of day it is had been used in prediction of interruptibility of office workers in the studies by Hudson et al. (2003) and Fogarty et al. (2005).

2.5.2 Interventions

Another application of affective computing is using interventions in human-computer interaction. Interventions can be used to study the detection of affective states, as seen in previous subsection, but also for behaviour change (Michie, Van Stralen, & West, 2011). An example in the usage of interventions is by Partala and Surakka (2004), who studied positive and negative interventions made with a speech synthesizer during a human-computer interaction task and observed better performance in the task after positive interventions.

There have also been interventions related to stress. MacLean, Roseway, and Czerwinski (2013) introduced wearable technology that mirrors a user's stress state in the form of a butterfly. While it was able to improve user performance in driving tasks, the authors also note that it became a stressor in itself for the users. Another challenge

was privacy, as most users of the technology stated they were not comfortable exposing this information about their affective state to other people.

Paredes and Chan (2011) investigated the effects of haptic feedback, games, and social networks on mobile phones used in interventions on stress relief. The authors emphasize the importance of timeliness and the type of intervention.

Later work by partly the same authors attempts to solve these problems in the mobile context. Paredes et al. (2014) use machine learning for task recommendation, which takes into account the individual in personality, coping skills and other ways, and the temporal circumstances such as time of the day and GPS location. Used interventions are re-purposed popular applications and websites, such as reading good news from the internet. Participants in a user study reported higher self-awareness of stress, lower depression-related symptoms, and having learned new ways of dealing with stress.

Haptic feedback has also been used in smartwatches for interventions. Costa, Guimbretière, Jung, and Choudhury (2019) have presented BoostMeUp, and it is based on the observation that people tend to associate external signals resembling heart rates as their own. In their experiment, slow haptic feedback was associated with better performance in a math task and fast haptic feedback with worse performance in the math task.

3 Methodology

This chapter documents the applied research methods. The section starts by describing the whole research process, and then goes one by one over the used research methodology: first giving definitions and then the specifics of the application of the methodology. More precise information on the details of the application of the methods can be found in the original publications.

3.1 Research process

The research methodologies used can be divided into two lines or branches of enquiry. Secondary studies were conducted for research questions **RQ1** and **RQ2**, while primary studies were conducted with a field study and repository mining to answer **RQ3** and **RQ4**. An overview of the research methodologies and data gathering techniques used is shown in Figure 2. Work advanced along these branches simultaneously, with the field study taking over priority when communication or activities with the study site were needed.

3.2 Reviews

Reviews are secondary studies, which can provide overviews of research areas, as well as provide summaries of specific research questions and hypotheses. Review-style articles can be divided into various sub-categories, such as bibliometric studies, systematic mapping studies and systematic literature reviews.

3.2.1 *Bibliometric studies complemented with natural language processing — Papers I and II*

Bibliometric studies assess research in a given field using reference information, and also the term scientometrics and infometrics are used according to Andrés (2009). We complemented this information with natural language processing (NLP) techniques topic analysis and text clustering. Topic modelling identifies the topic of the text and possibly how it changes throughout the document (Li & Yamanishi, 2003). According to Beil, Ester, and Xu (2002), text clustering structures large sets of data, and in our case based on the identified topic. This was done as the acquired data sets were large in both cases, and more conventional reviews would have been too laborious.

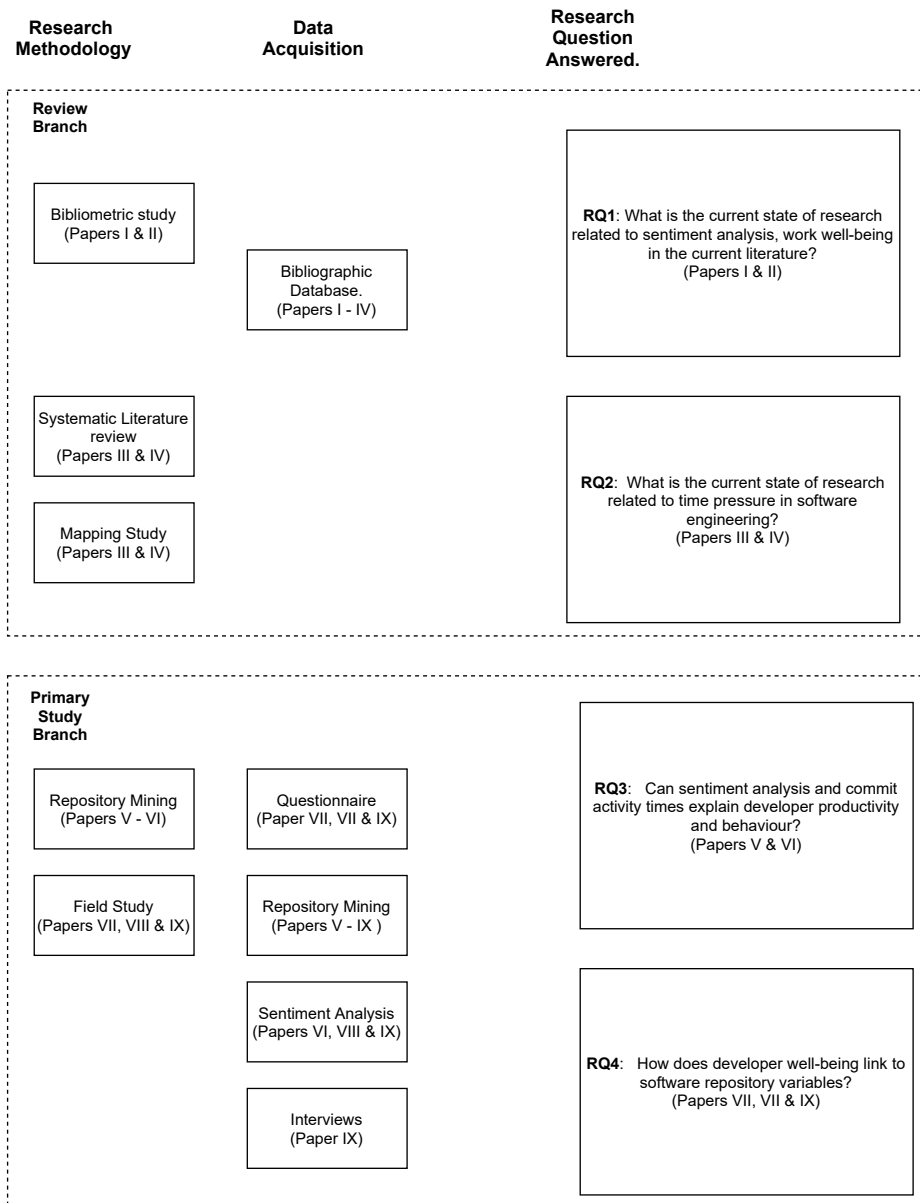


Fig. 2. Mapping of publications to used research methodologies and research questions.

Table 6. Characteristics of Papers I and II.

Feature	Paper I	Paper II
Search terms:	“time pressure”, “deadline pressure”, “schedule pressure”, “time budget pressure”, “deadline pressure”, “pressure of time”, “pressure of schedule”, “pressure of time budget”, “pressure of deadline”, “speed-accuracy tradeoff”.	“sentiment analysis”, “opinion mining”, “sentiment classification”, “opinion analysis”, “semantic orientation”, sentiwordnet, “opinion classification”, “sentiment mining”, “subjectivity analysis”, sentic, “subjectivity classification”.
Number of sources	1270	6996
Number of identified topics	79	108

To answer **RQ1**, we performed two studies combining aspects of bibliometric studies and literature reviews. Both of these studies started with determining search strings by familiarizing oneself with different synonyms related to the area of study. In practice, this meant adding synonyms to the search string. In Paper I terms such as “deadline pressure”, “time budget pressure” and “schedule pressure” were used in addition to “time pressure”. In Paper II, terms such as “opinion mining”, “sentiment classification” and “opinion analysis” were used in addition to “sentiment analysis”. The searches for these paper were performed with the search engine Scopus. The acquired data-sets included 1,270 papers in Paper I’s case, and 6,996 scientific papers in Paper II’s case. These are shown in Table 6.

After this, in both studies, Latent Dirichlet Allocation (LDA) was used to cluster the found papers. LDA is a soft clustering algorithm, meaning it assumes texts can have multiple topics inside them. In practice, this means that instead of a paper being assigned to a single topic or "cluster", it can be part of multiple topics. An example of this could be a paper studying usage of software systems under time pressure in a healthcare context, which would be placed in theory in both software and healthcare topics. The origins and recommendations for using this methodology are in the paper by Griffiths and Steyvers (2004), whose recommendations for hyper-parameter settings were followed in both papers (i.e., $\alpha=50/k$ and $\beta=0.1$). These parameters describe how likely each document is to contain multiple topics (alpha) and how likely each topic is to contain multiple words (beta).

The LDA topics were further analyzed iteratively with qualitative coding (Patton, 1990), and the coding schemes are presented in the publications. The Mindmup¹ tool

¹<https://www.mindmup.com/>

was used for collaboration between authors and to visualize these coding schemes. Additionally, the LDA topics were further visualized with wordclouds, and the R package wordcloud was used for this (Fellows, 2012).

3.2.2 Systematic maps and reviews — Papers III and IV

According to B. Kitchenham and Charters (2007), a systematic literature review (SLR) is defined as “a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest”. Similarly, systematic mapping studies (SMS) are reviews that provide more coarse-grained overviews of research areas (Petersen, Feldt, Mujtaba, & Mattsson, 2008). Moreover, systematic maps provided by systematic mapping studies categorize research reports and results, often in a visual form.

To answer **RQ2**, we performed a study combining parts of both SMS’s and SLR’s. Paper III is the study in its entirety, and paper IV summarizes and visualises the results of the study and discusses the practical implications of the work in a scientific magazine. For these studies, we combined SMS and SLR methodology by systematically mapping and exploring the larger body of knowledge about time pressure in software engineering. This was done by taking a subset of the acquired sources with which a specific SLR style research question on the effects of time pressure was answered.

Figure 3 shows a flow chart of the methodology used for Paper III. The initial literature search included the search made for Paper I with Scopus, additional searches with the keyword "schedule compression" in Scopus, and additional searches with the same keywords using Google Scholar. Later, the searches were complemented with snowballing. In snowballing, we followed Wohlin’s snowballing guidelines (Wohlin, 2014). This meant defining a set of seed papers from included papers, and then searching for papers that were cited by the seed papers, as well as papers that cited the seed papers using Google Scholar. Additionally, we also searched all the papers by the authors of the seed papers.

Inclusion and exclusion criteria were then used to filter these search results. Criteria were constructed iteratively when familiarizing oneself with the first searches. Table 7 shows the inclusion and exclusion criteria used.

The selection process started with the author of this thesis reading the title and the abstract of the results. The exclusion criteria E1 and E2 were used to exclude non-scientific sources and papers not written in English. Reading the paper would continue until a definitive decision could be made. To be included with inclusion criteria I2 the paper must contain empirical evidence related to time pressure in software

Table 7. Used inclusion and exclusion criteria for Paper III. (Reprinted, with permission. © 2020 Elsevier).

Inclusion / Exclusion	Criteria
Inclusion 1	The main focus is time pressure in software engineering.
Inclusion 2	The paper presents empirical evidence of time pressure in software engineering.
Exclusion 1	The paper was written in language other than English.
Exclusion 2	Not a scientific source.
Exclusion 3	The task studied is not a software development task.

engineering. Theoretical papers without their own empirical evidence were included with inclusion criteria I1. These inclusion criterion were also used qualitatively in judging what constitute empirical evidence: after the fact explanations and brief mentions without concrete evidence gathering were not included.

Data extraction from the included sources was done iteratively with NVivo². First, the coding scheme included only a few codes, such as *process phases and causes of time pressure*. However, as sources were coded, the scheme was iteratively improved. Once the coding scheme was complete and all the studies were coded in the first iteration, the coding scheme was checked by multiple authors. After this, all included studies were coded a second time to ensure that the same coding scheme was used on all the sources. The resulting coding scheme, as well as results of repository searches, verdicts of inclusion and exclusion criteria and other material, can be found in a replication package³.

3.3 Primary study branch

Primary data has been defined as “data that are collected for the specific problem at hand, using the procedures that fit the research problem best” by Hox and Boeije (2005). Hence, primary analysis has been described as the original analysis of the data for a research study (Glass, 1976). In this thesis, this branch consists of five different papers, and this section describes their data gathering and analysis techniques.

²<https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home>

³<https://figshare.com/s/0662c66e0705ebf8dca7>

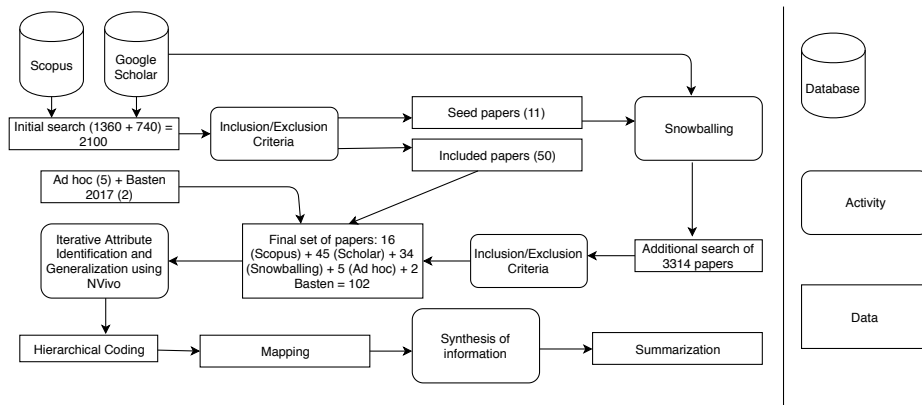


Fig. 3. Flow chart of research methodology for Paper III. (Reprinted with permission. © 2020 Elsevier).

3.3.1 Repository mining — Papers V and VI

Repository mining at scale — Paper V

Paper V investigates empirical measures of software engineers' work patterns by studying the time stamps of commits on a large scale. Time stamps can act as a proxy for working times, as they indicate completion time of work units done on the project. The data was acquired from the Git and Mercurial repositories of Mozilla and Apache, as well as a local company's git repository. The local company project is the same one that provided data for the field study used for papers VII, VIII and IX. GrimoireLab, by Dueñas, Cosentino, Robles, and Gonzalez-Barahona (2018), was used for mining the data. Thus, this data set includes time stamps of commits from 86 large open source projects with both paid and unpaid developers, as well one industrial project with solely paid developers. In total, this resulted in 451,116 commits.

To study the effect of demographics on commit patterns, background information from the top 10% most active developers to the Mozilla Firefox project in terms of commits. The data was manually gathered from publicly available sources, including LinkedIn, developers' personal websites, and other such publicly available CVs. This resulted in a dataset of 278 developers (out of 2,755), each with 147 or more commits. Out of these, 227 are paid and 48 are volunteers. The final dataset includes information regarding job status (paid or unpaid), location, experience in the software industry, and the position at Mozilla.

The acquired data of commit time stamps was visualized with barplots, histograms, lineplots, k-means clustering, comparison word-clouds, and comparing commit message

bi- and trigrams during and outside office hours. Three clusters of work patterns were found with k-means clustering, and gathered background information was used to provide insights about these clusters. Illustrating the clusters was done by plotting their share of commits within office hours and during abnormal hours. Additionally, the effect of demographic information is studied by looking at the median percentage of commits made within office hours per background characteristic.

Repository mining at depth — Paper VI

Paper VI examines possible links between developer sentiment in instant messaging and activity in the code repository. We had access to the project’s Git repository and acquired metrics related to productivity: number of commits and number of lines changed per day. While these metrics were available for a larger period, we collected the data for the days the questionnaire was run, meaning for a period of eight months for each developer.

The company provided us with chat logs of their developer chat room from slack and hipchat, where the language used in the chatrooms was Finnish. We translated one lexicon used in the software engineering context for measuring arousal (Mäntylä et al., 2017) and one for valence to Finnish for sentiment analysis purposes. The chat logs were lemmatized using the open-source software Voikko (Pitkänen, 2012), and then each lemma was scored for valence and arousal using the translated lexicons. This resulted in variables *negative valence*, *positive valence*, *low arousal*, *high arousal*, *minimum valence*, *maximum valence*, *minimum arousal*, and *maximum arousal*. We also took into account emojis and emoticons, which are “picture characters” or pictographs (Miller et al., 2016). These we manually classified according to Plutchik’s wheel of emotions (Plutchik, 1991): joy, sadness, surprise, confusion, and anger. For each developer, we calculated the percentage of messages containing *emoticons* and emoji, *emoticons and emoji related to joy*, *emoticons and emoji related to surprise, sadness, and confusion*. Due to the low number of emoticons and emoji for the latter group of emotions, we combined them in one variable named *sadconfusionsurprise-emo*. These same variables were later used in Paper IX and are shown in Figure 4.

These variables were analyzed by building different regression models. First, possible links between productivity and sentiment were investigated by making *pairwise* regression models, where the log of number of commits or number of lines changed were predicted with variables expressing sentiment. Afterwards, *step-wise* builds predicting productivity with regression models were done using Akaike’s information criterion (see Akaike (1998)), which is an estimate of prediction error that includes a penalty for increasing the number predictors for a given model. In practice, this meant adding

sentiment variables as predictors, which produced the lowest AIC score, and stopping once AIC score did not decrease.

3.3.2 Field studies — Papers VII, VIII and IX

As we were completing the analysis for Paper V and looking at developer commit activity timestamps qualitatively, we realized that identifying time pressure was really challenging. We could not observe clearly bigger commit activity close to releases, nor could we observe clearly increased activity outside office hours and during the weekends. Thus, we came to the conclusion that we needed to ask the developers how they felt to know when they were under time pressure.

According to Courage and Baxter (2005), field studies refer to “broad range of data gathering techniques at the user’s location — including observation, apprenticeship, and interviewing.” In our studies, the data gathering at the user’s location included running a daily questionnaire for software developers, using repository mining to acquire data related to productivity, job events, and chat activity, and doing sentiment analysis on the expressed sentiment in the chatlogs. Lastly, interviews were done to contextualize and explain the results after the quantitative data were analyzed.

Questionnaire — Experience Sampling — Papers VII, VIII and IX

Experience sampling methodology (ESM) refers to a research procedure where people are asked how they feel, think, or do during their daily lives (Larson & Csikszentmihalyi, 2014), sometimes even physiological data can be gathered (Alliger & Williams, 1993). Gathering data by asking respondents for their experiences can be done in a variety of ways, including pen and paper, beepers, wrist watches, email or mobile devices (Scollon, Prieto, & Diener, 2009) (Van Berkel, Ferreira, & Kostakos, 2017). In the context of our study, the strengths of ESM are ecological validity, reduction of memory bias and its usage together with other research methodologies (Scollon et al., 2009), which follow from the recording of "real life" experiences outside of a lab.

To record the daily experiences of software developers related to their well-being, we constructed a questionnaire from the survey done by Elovainio et al. (2015). In the end, we decided to include five single items that measure variables related to job well-being. In addition, we added a single item specific to software engineering. We opted for single-item measurements because we intended to use the questionnaire in an experience sampling study, meaning the questionnaire was to be taken daily. Thus, requiring the developers to answer dozens of questions daily would not have been practical or possible.

Though the items were picked from the survey by Elovainio et al. (2015), the items have their origins from other questionnaires. Items about independence and interruptions are from Karasek's Job Content Questionnaire (Karasek et al., 1998); the item measuring Hurry is from the Harris stress index (Harris, 1989); the item referring to distress is originally from the general health questionnaire "GHQ-12" (Goldberg & Blackwell, 1970); and lastly, the question concerning sleeping problems is from the Jenkins scale (Jenkins, Stanton, Niemczyk, & Rose, 1988). These questions were slightly modified to fit our five-point scale by asking the respondents to rate these six items with the question: "How frequently has the following condition occurred since the last time you answered this survey?". From 1 to 5, the corresponding textual answers were "very rarely or never", "rarely", "once in a while", "often" and "frequently or continuously".

Before starting the data collection with the questionnaire, first the questionnaire was trialed by the authors of the study. After practical matters in running the questionnaire with Webropol were solved, the purpose of the study was explained to the project personnel in a meeting where they could also ask questions.

Repository Data Gathered and the Measurement Model used in Papers VI, VIII, IX

For Paper VIII, only a simple count variable was used related to chat messages. This means the number of chat messages sent by the developer to measure chat activity. Productivity metrics were also used; this included number of commits, number of code lines changed, and the number of files changed.

For paper VI and IX, we used the same sentiment analysis variables that are presented in Section 3.3.1. Exploratory factor analysis, used in paper IX, is used to study the structure of the underlying variables in general (Thompson, 2004). In practice, this means using multiple variables, such as commits, lines of code changed, and the number of files changed to measure one underlying construct. The optimal number of underlying factors was calculated using the function `fa.parallel`⁴, after which the function `fa`⁵ was used to calculate the minimum residual (minres) solution using 100 iterations. Both of these functions are from the "psych" package for R (Revelle, 2011).

Our measurement model (Figure 4) shows the relationships between latent variables and their indicators (Bollen, 2001). The results of exploratory factor analysis can be seen in the measurement model, specifically on the left side of Figure 4. On the right side, correlations between questionnaire variables are shown. Oval shapes under repositories

⁴<https://www.rdocumentation.org/packages/psych/versions/2.0.8/topics/fa.parallel>

⁵<https://www.rdocumentation.org/packages/psych/versions/2.0.8/topics/fa>

mark the factors acquired with exploratory factor analysis, and the rectangles mark variables both from the repositories used for factor analysis, and the variables acquired with the questionnaire. Lines between variables and factors show weights, with dotted lines signaling negative weights.

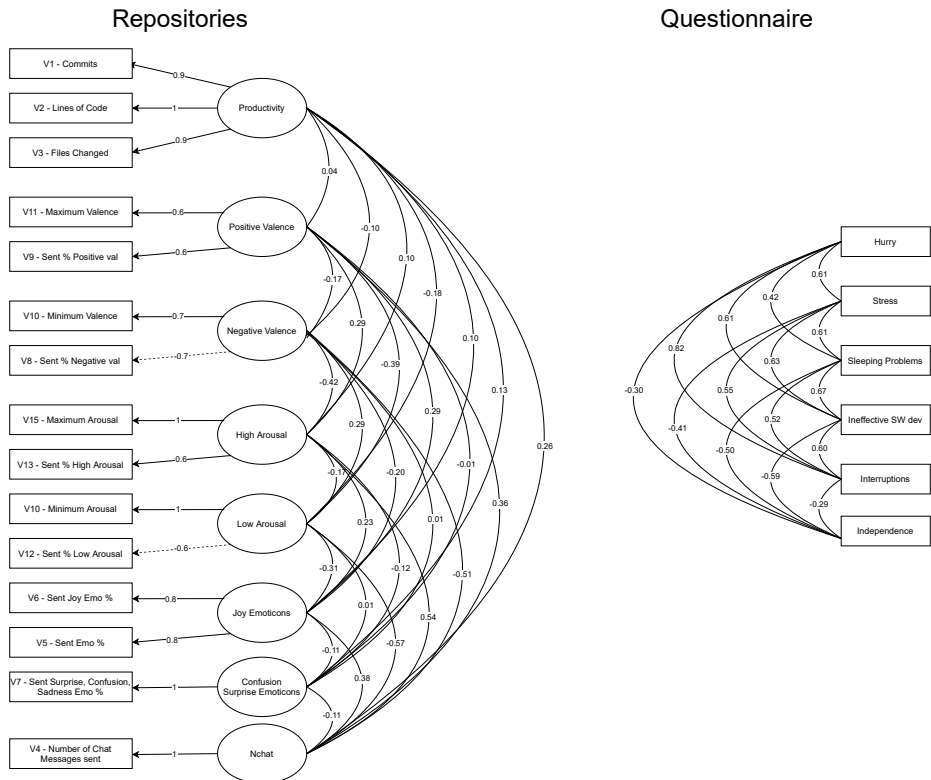


Fig. 4. Measurement model of Paper IX: variable weights to factors and correlations between latent variables resulting from a factor analysis. (Under CC BY 4.0 license from Paper IX © 2021 Authors).

Interviews — Paper IX

The quantitative data was complemented with semi-structure interviews for paper IX. Semi-structured interviews have been recommended as a supplement to questionnaires when specifically important questions remain after collecting quantitative data (Adams, 2015). Interview questions were drafted collectively by the authors of the paper, aiming

for open-ended questions for which follow-up questions and clarifications could be asked. These questions can be found in a Github repository⁶. Once interviews were done, they were transcribed verbatim.

Analysis of transcripts was done following the strategy of Schmidt (2004). This meant repeated reading of the transcripts and a development of the simple coding scheme. Three codes were used: (1) facilitating activities helping well-being; (2) barriers to well-being; and (3) explanation of quantitative results. The next phase of the analysis included quantifying codes and finding uniform and coherent codes across interviewees.

Model construction and evaluation Paper VII

Psychological networks are used to visualize potential network structures of psychological and other components (Epskamp, Borsboom, & Fried, 2018). One such tool is the Gaussian graphical model by Epskamp, Waldorp, Möttus, and Borsboom (2018), which “estimates a network of partial correlation coefficients—the correlation between two variables after conditioning on all other variables in the data set”. These networks consist of nodes representing observed variables, and edges or lines showing the observed relationships between the variables (Epskamp, Borsboom, & Fried, 2018). In our case, both temporal and contemporaneous networks were evaluated. Temporal network refers to a within-subject network that shows temporal prediction, and contemporaneous network referring to a within-subject unidirectional network which shows the effects after taking the temporal effects into account (Epskamp, Borsboom, & Fried, 2018). Gaussian graphical models have been advocated for exploratory studies of data sets (Bhushan et al., 2019).

The resulting time series from the questionnaire were also analyzed with Spearman correlations. The resulting time series from the questionnaire were tested with the Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests. ADF rejected the null hypothesis of non-stationarity, and KPSS could not reject the null hypothesis of stationarity. Thus, stationarity was assumed.

For the purposes of Paper VII, missing values from days the questionnaire was not answered were handled by carrying back values. This was argued for with the way the questions were asked: “How frequently has the following condition occurred since the last time you answered this survey?”. This was not done in Papers VIII and IX.

⁶<https://github.com/M3S0ulu/autotime-esm>

Model construction and evaluation Paper VIII

Analysis of the acquired experience sampling data together with data acquired from repositories was done with multiple binary logistic regression in Paper VIII. Logistic regression models have a binary dependent variable, while predictors can include continuous, categorical, or binary variables (Hilbe, 2009). Its usage has been advocated in the social sciences by King (2008). Splitting the data to binary form was already popular and criticized in the 90's by Maxwell and Delaney (1993), though defenses of the practice have been published in consumer psychology in more recent times (Iacobucci, Posavac, Kardes, Schneider, & Popovich, 2015a, 2015b).

For the purposes of paper VIII, the questionnaire variables were transformed to binary form, and repository data in number of commits, lines of code changed, and the number of chat messages sent. Both were used in binary form when used as a dependent variable, though repository variables were only transformed to logarithmic scale when used as predictors. Splitting the data was not done in Paper IX. Transformation into logistic form was done using R's cut function⁷. This was argued for by personalizing the scales for each individual, *i.e.*, one developer's hurry of 3 ("once in a while") might be another's 2 ("rarely"). Various measures for model performance were given, including the area under the ROC curve (Bradley, 1997) and F1-score, as well as statistical significance for predictor variables in the models. We used control variables for both weekly seasonality and autocorrelation. These control variables have been advocated for by West and Hepworth (1991).

Model construction and evaluation Paper IX

We used generalized linear mixed effects models in Paper IX. Mixed effects models allow for modeling data that have a grouped structure, such as repeated measures in a longitudinal setting (Faraway, 2016). For generalized mixed effects models, we calculated marginal and conditional R^2 values. Marginal R^2 value represents the variance explained by fixed effects, and conditional R^2 value is the variance explained by the entire model. Calculating R^2 values for mixed effects models is based on the work by Nakagawa and Schielzeth (2013); moreover, we used the MuMIn implementation by Barton (2009) to calculate it.

⁷<https://www.rdocumentation.org/packages/base/versions/3.5.0/topics/cut>

4 Original research papers

This chapter presents and summarizes the individual studies of which thesis is composed. All of the papers have been published in peer-reviewed venues. The quality of publication venues is classified by the Finnish Publication Forum⁸. Publication venues are ranked to categories in the JUFO levels: not meeting scientific requirements, basic level, leading level, and highest level, where the highest level corresponds to level 3, and the level 0 to not meeting scientific requirements.

Quality assessment of publication venues is also done by the Computing Research and Education Association of Australasia, CORE Inc⁹. CORE rankings include ranks of A*, A, B, C, and other, where A* corresponds to the highest level. As of making this thesis in 2021, ranks of A* and A comprise 23% of the top conference venues¹⁰ in computer science, and 16% of the top journal venues¹¹ in computer science. Table 8 presents the papers, publication venues, publication types, JUFO-rankings, CORE rankings, and publication years. The abbreviation SEmotion in Table 8 refers to International Workshop on Emotion Awareness in Software Engineering.

Next, for each individual study, the research goals and results are provided. The contribution of the author of this thesis is also mentioned. The previous section 3 already presented the details of the research methodology used. The answers to the research questions and more additional discussion are provided next in Section 5.

4.1 RQ1: What is the current state of research related to sentiment analysis and time pressure in the current literature?

4.1.1 Introduction

The first research question is answered with two high-level review articles, as the topics of sentiment analysis and time pressure are very broad. Paper I deals with time pressure, while Paper II deals with sentiment analysis. Both of these articles use text clustering to form clusters of papers, which are then analyzed with qualitative coding and word-clouds. The methodology was introduced in subsection 3.2.1.

⁸<https://julkaisuforum.fi/en>

⁹<https://www.core.edu.au/home>

¹⁰<http://portal.core.edu.au/conf-ranks/>

¹¹<http://portal.core.edu.au/jnl-ranks/>

Table 8. Information related to publications, venues, JUFO level, CORE ranking and publication type.

Paper	Publication Venue	Year	JUFO level	CORE Ranking	Publication Type
I	SEmotion	2017	-	-	Workshop
II	Computer Science Review	2018	1	-	Journal
III	Information and Software Technology	2020	3	A	Journal
IV	IEEE Software	2021	2	B	Magazine
V	International Conference on Software Engineering	2018	2	A*	Conference
VI	SEmotion	2020	-	-	Workshop
VII	SEmotion	2018	-	-	Workshop
VIII	International Symposium on Empirical Software Engineering and Measurement	2018	2	A	Conference
IX	Empirical Software Engineering	2021	3	A	Journal

4.1.2 Paper I: Reviewing literature on time pressure in software engineering and related professions: computer assisted interdisciplinary literature review.

Goal and motivation: The goal of the paper was to explore and analyze literature related to time pressure and synthesise knowledge based on this literature. The literature search yielded 1270 papers, which were analyzed with text clustering based on abstracts. The log-likelihood measure produced 79 as the optimal number of topics. These were qualitatively put under nodes using a tool called FreeMind. Figure 5 shows a reproduction of the highest level of qualitative coding, with the number of topics added under each node. As seen, time pressure has been studied in a wide variety of occupations, with various tasks, in relation to well-being, demographics, communication, safety and commerce.

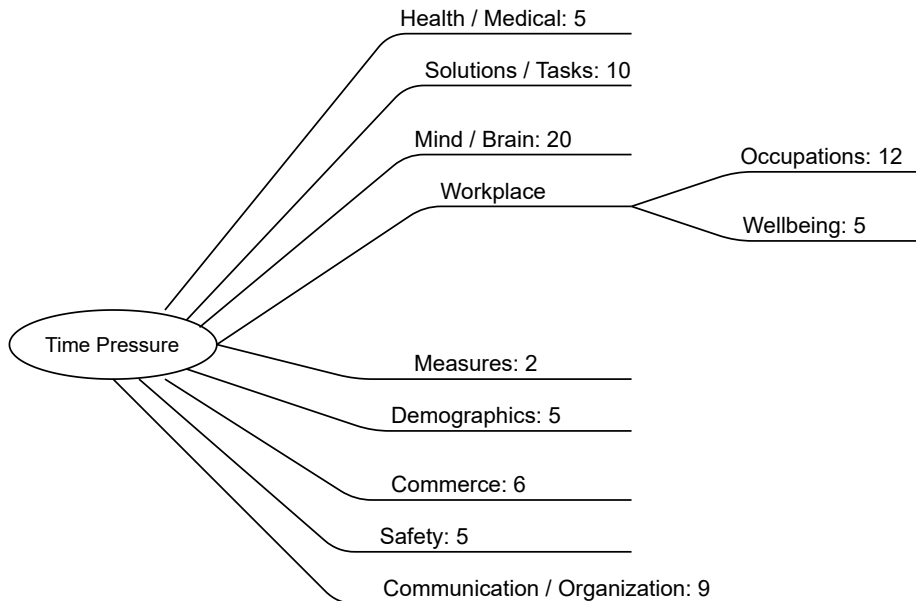


Fig. 5. Top level of the classification tree for Paper I. (Reprinted, with permission, from Paper I © 2017 IEEE).

Results: Word clouds were used to visualize the contents of four topics, three of which are under the node "Occupations" in Figure 5 and one under the node *well-being*. The topics, named after the most commonly used words, were *software engineering*, *project work*, *auditors and auditing* in the node under occupations, and *job satisfaction* in the node *well-being*. Furthermore, the scattered nature of the main nodes in Figure. 5 shows how broadly time pressure has been studied.

By examining the most relevant clusters in relation to software engineering, a list of testable hypotheses were inferred from some of the reviewed papers. Sources from auditing especially analyzed for possible hypotheses due to some similar characteristics to software development. Examining papers under the topic *job satisfaction* with word-clouds unveiled theoretical frameworks from occupational psychology; papers cited the job demands-resources model (Bakker & Demerouti, 2007; Karasek & Theorell, 1990) as an explanation for employee well-being.

The majority of software papers were related to software testing and quality assurance, hinting at the possibility that the effects of time pressure are most common there. Additionally, performing this analysis provided a set of papers that could be extended to a systematic mapping and a systematic literature review.

Author contribution: The author of this doctoral thesis was the first author, and had the main role in doing the analysis and writing the paper.

4.1.3 Paper II: The evolution of sentiment analysis — A review of research topics, venues, and top cited papers.

Goal and motivation: The goal of the study is to provide an overview of the field of sentiment analysis with a computer-assisted literature review using text mining, citation analysis, and qualitative coding. This bibliographic study analyzes 6,996 papers related to sentiment analysis acquired from the Scopus and Google Scholar databases.

Results: The main finding of the paper is the near 50-fold increase in the decade from 2005 to 2015 of the publications of sentiment analysis studies. Studies are also found to be published in varying venues, with the top 15 publication venues only corresponding to 30% of publications. In total, 108 topics were discovered with clustering, which shows the wide range of topics related to sentiment analysis.

Regarding the roots of sentiment analysis, some of the earliest articles found included manual efforts in capturing public sentiment with questionnaires from the first decades of the twentieth century (*e.g.*, Droba (1931)). The breakthrough of modern sentiment analysis came with machine learning methods, the internet, and vast amounts of analyzable data: the top cited paper by Pang and Lee (2008) is from 2008, and it focuses on product reviews available on the Web.

The wide range of topics is further shown in the top level of the qualitative classification tree in Figure 6. To demonstrate the wide variety of topics, taking a look at the application domain is enough. It is divided into 7 nodes, under which lies a total of 41 topics. These include *society*, *security*, *travel*, *finance*, *medical*, *entertainment* and lastly the node *other* collecting topics related to application domains not fitting to aforementioned topics.

Author contribution: The author of this thesis was the third author, responsible for reading and summarizing the most cited papers presented in Sections 3.6.1 to 3.6.4 in the original publication. The author was also part of the overall article writing process, making contributions to the introduction, discussion and conclusions.

4.1.4 Main contributions and findings

- List of hypotheses inferred from time pressure-related literature that might help in detection of time pressure in software engineering.
- Nearly fifty-fold increase in the number of publications related to sentiment analysis.

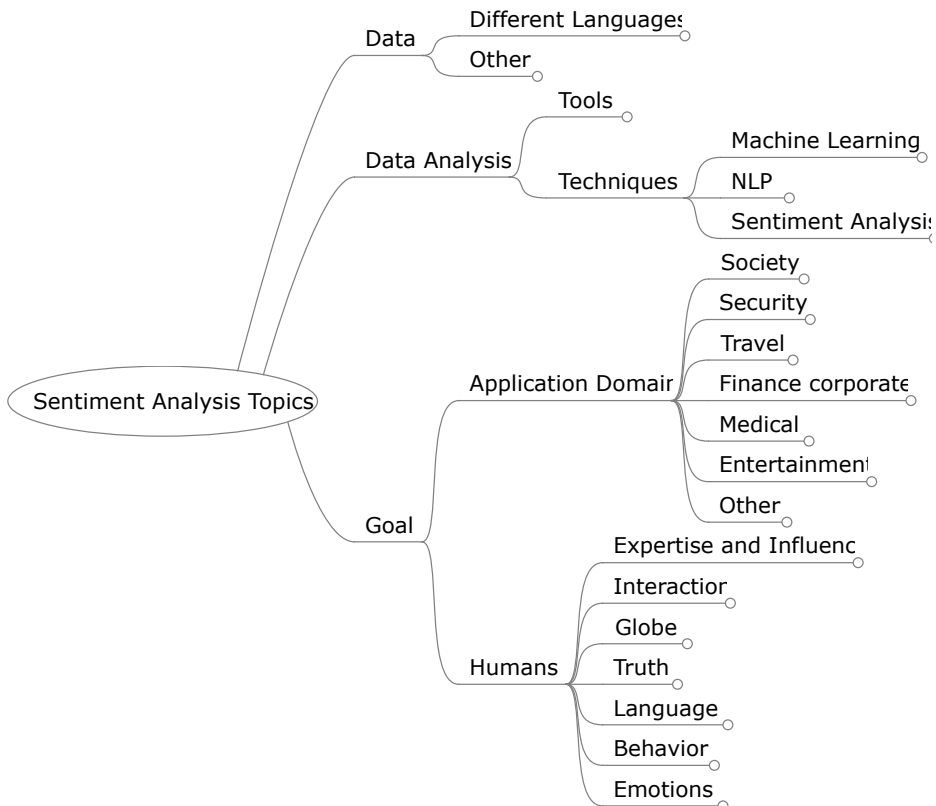


Fig. 6. Top level of the classification tree for Paper II. (Reprinted, with permission, from Paper II © 2018 Elsevier).

4.2 RQ2: What is the current state of research related to time pressure in software engineering?

4.2.1 Introduction

To answer this research question, a study combining aspects of mapping studies and systematic literature reviews was conducted. Unlike RQ1, RQ2 focuses specifically on the software engineering context. Paper III presents this study, while Paper IV summarizes in a shorter form for practitioner audiences. The methodologies used for these studies are introduced in detail in subsection 3.2.2.

Table 9. Found sources by investigated software development process phases and methodologies. (Reprinted, with permission, from Paper III. © 2020 Elsevier).

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

4.2.2 Paper III: Time pressure in software engineering: a systematic review.

Goal and motivation: This systematic review extended Paper I, and analyzed 102 sources in total. It built a systematic map of studies related to time pressure in the software engineering context and a systematic literature review for a subset of the found papers, which were published in a single review paper. The goal of the study is to provide an overview of the causes and effects of time pressure in software engineering literature. Specifically, this includes existing definitions of time pressure, metrics used to measure time pressure, and mapping of included studies to software processes and approaches. We also synthesized the outcomes of found quantitative empirical studies in a systematic literature review and gave practitioner takeaways based on the analyzed material.

Table 10. Effects of time pressure. (Modified slightly from Paper IV. © 2021 IEEE).

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

Results: One result of the study was identifying and presenting used metrics and measurements of time pressure in software engineering context, as shown in Tables 1 and 2. Common metrics include different ratios between estimates and results, such as customer negotiated time, actual schedule, and remaining effort. In experiments, time limits, different task difficulties, and rewards for faster completion have been used to create time pressure.

The included studies in the review were mapped to the process phases of the waterfall model (Sommerville, 1996). Categories of the waterfall model were complemented with two other categories. Category whole process or approach refers to papers covering multiple process phases. Category others covers papers that did not map into process phases, but where multiple papers concentrated on a single theme (*e.g.*, detection of time pressure). Table 9 shows the number of studies found for each category. As can be seen: most commonly studies focus on quality assurance, cost estimation, and process simulation. Other bibliographical information was also examined, such as the publication years and the most common publication venues of the selected studies, which can be found in the original paper.

The study also systemically extracted causes of time pressure from the found literature. These were generalized into errors in effort estimation or project management and company culture. Hence, the causes of time pressure can be technical, social, or a combination of these. Poor scheduling and management can be caused by a lack of historical data for effort estimates, lack of buffer time for any changes during the development, as well as caving in for business pressures and promising delivery in an unrealistic schedule. However, a multitude of sources tell a story of long hours and “crisis mentality” that are part of the company culture. This can include demands for prioritization of work over private lives as a way of career advancement, focusing on individual heroics over development process, promoting a sense of urgency without rest and refocus periods, and specialized roles and personal modes of coordination.

The study also found studies concentrating on the effects on individuals. Multiple studies linked time pressure to negative effects on individual well-being, specifically

in software development context. In a survey, time pressure was seen as the most frequent external cause of unhappiness. These include decrease in confidence, increase in feeling of uneasiness, willingness to postpone decisions and to re-use code unethically. However, perhaps most importantly, time pressure is also linked to burnouts and depressive symptoms, which in turn have been linked to clinical depression. Three studies were also found reporting a mediating effect of agile ways of working on the effects of time pressure. This included decrease in overtime work, increased empowerment leading to better ability to deal with stress, and reporting of less feelings of job strain.

Table 10 summarizes the results of empirical studies and the assumptions underlying software process and estimation models on the effects of time pressure on software development efficiency and quality. The observed effects include increased or decreased efficiency, or an inverted U-shaped relationship. The inverted U-shaped relationship means that time pressure increases efficiency up to certain point, but after that, increases in time pressure cause a decrease in efficiency. 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 (see Section 2.1), and two papers report either a decrease or no effect on efficiency. The context or the nature of the study may create a situation in which time pressure has no effect on efficiency, *i.e.*, there is a point after which one cannot go faster.

Surprisingly, decreased efficiency under time pressure is assumed for eight different cost estimation and process simulation models, which is in conflict with empirical studies. An increase in efficiency is the underlying assumption in 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. This difference in assumptions between models was already noted in 1992 by B. A. Kitchenham (1992). We suspect that different time scales play a role here in some of the differing results between empirical studies and software process and simulation models.

In Table 10, nine empirical studies support a reduction in software quality due to time pressure, while four studies report the opposite. The four latter studies had either low statistical significance or achieved statistically significant correlation, but did not use more robust methodology in multiple regression, making the finding less reliable. Cost estimation and scheduling models unanimously support a decrease in quality under time pressure (schedule compression).

Author contribution: The author of this thesis is the first author, and was responsible for applying exclusion criteria to 5,414 sources found with repository searches and

snowballing, developing a coding scheme and the usage of it for the included sources, and had the main role in writing the review.

4.2.3 Paper IV: What do we know about time pressure in software development?

Goal and motivation: This paper presents a practitioner-oriented aggregation and summary of Paper III. The rationale for writing a summary is the fact that the original paper was published in an academic venue targeting mostly the research community. However, we believe that giving a practitioner-oriented summary for industry based on the results would be valuable.

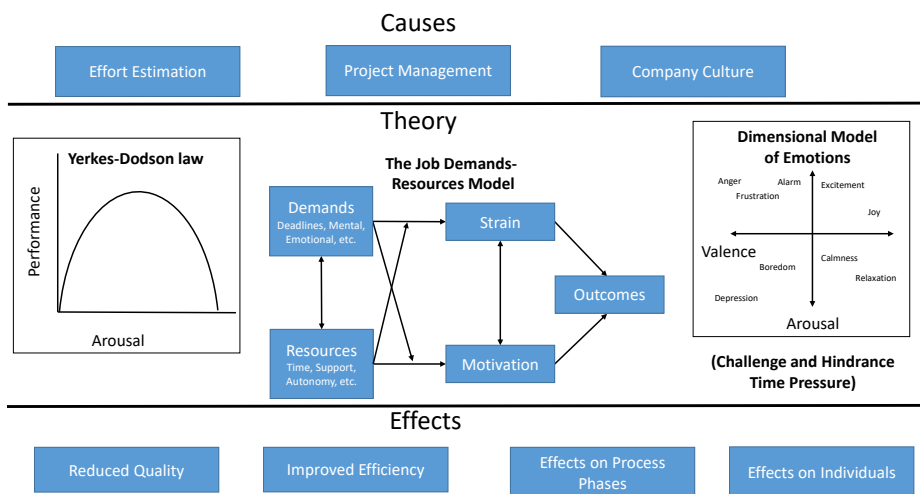


Fig. 7. Causes, theory, and effects of time pressure in the context of software engineering. (Reprinted, with permission, from Paper IV © 2021 IEEE)

Results: The main contribution of the paper is Figure 7 summarizing theories related to time pressure from fields other than in software engineering, and tying these to the causes, theories, and observed effects of time pressure found in software engineering literature. Longer explanations for all parts of this figure exist in this thesis: for cause and effects in the previous Section 4.2.2, and for theories Section 2. For conciseness, they are not repeated entirely here. Specifically, the Yerkes Dodson law and challenge-hindrance framework are presented in Section 2.1, and the demands-resources model subsection in 2.1. The challenge hindrance time pressure framework by Chong et al. (2011) is illustrated with the dimensional model of emotions in Figure 7.

Figure starts with introducing the different kinds of causes of time pressure: poor effort estimates, poor management and company culture fostering time pressure. Different theories related to the effects of time pressure are then introduced, highlighting the U-shaped relationship between pressure and performance by the Yerkes Dodson law, and job strain being the outcome of imbalance between demands and resources dictated by the JD-R model. The challenge hindrance framework LePine et al. (2005) is illustrated with a dimensional model of emotions, where we placed both challenge and hindrance time pressure on the higher arousal axis, but on the opposite sides of the valence axis.

Lastly, the paper focuses on the practitioner takeaways from Paper III. These included the quality trade-off with time pressure being worse in tasks with an algorithmic nature, positive effects of intermediate deadline setting with agile methods, and the possibility of mediating the effect of knowledge on reduced quality.

Author contribution: The author of this thesis is the first author, and had the main role in summarizing the previous article and writing of the paper.

4.2.4 Main contributions and findings

- A set of definitions and metrics of time pressure is shown in Table 1.
- Tying of these definitions to theory in Yerkes-Dodson’s law and Challenge-Hindrance time pressure.
- A systematic map of studies related to time pressure in software engineering is shown in Table 9.
- A summary on the effects of time pressure on efficiency and quality in the software engineering, shown in Table 10. The majority of empirical studies support increased efficiency and reduced quality under time pressure.

4.3 RQ3: Can sentiment analysis and commit activity times explain developer productivity and behaviour?

4.3.1 Introduction

Two studies using repository mining were done for this research question. Paper V looks at developer behaviour on a large scale using mostly publicly available data in the form of commit times, whereas paper VI takes a deeper look at sentiment analysis of instant messaging in a single project. The research methodologies used are introduced in detail in subsections 3.3.1 and 3.3.1.

4.3.2 Paper V: Do programmers work at night or during the weekend?

Goal and motivation: The goal of this study was to examine the commit activities of software developers to establish baselines for different kinds of software projects. In total, the study consisted of 86 open source software projects, specifically in relation to outside office hours commit activities.

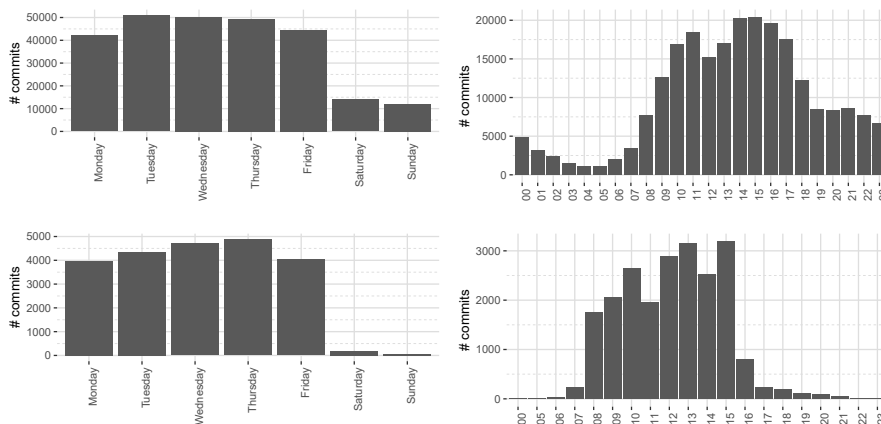


Fig. 8. Distribution of commits across hours at Mozilla and the local company. (Modified slightly from Paper V. © 2018 IEEE).

Results: Figure 8 shows commit patterns of developers from all Mozilla projects¹² and from a single project from a local company. The top left of the figure shows weekly commit patterns of all Mozilla projects, top right the daily hours during which commits were made in all Mozilla projects, with bottom left showing the weekly commit patterns and bottom right showing the daily commit patterns of the local company project. From these figures, it can be seen that commits follow a weekly rhythm, where most of the commits are made during the workweek and minority during the weekend. It can also be seen, that the commits follow a circadian rhythm: number of commits pick-up considerably after eight o'clock and wanes off in the evening. There is considerable difference between Mozilla and the local company project in the evening activity however.

Three work patterns for Mozilla Firefox developers who belong to the 10% commit authors were identified using k-means clustering, and these patterns are shown in Figure 9. The work week is here divided into 12 periods, and commits during office hours are seen as normal, and commits outside of office hours or during the weekend

¹²<https://hg.mozilla.org/>

Table 11. Paid and unpaid developers of Firefox inside three clusters based on working hours. Numbers respectively indicate the number of developers, the percentage of developers in a given cluster, and the total percentage from the same paid/unpaid status. (Reprinted, with permission, from Paper V. © 2018 IEEE).

Paid	Outside office hours cluster	Average cluster	Office hours cluster
No	65 (50% / 47.9%)	15 (11.6% / 31.3 %)	10 (10% / 20.8 %)
Yes	23 (50% / 10.1%)	114 (88.4% 50.2%)	90 (90% / 39.7%)

as abnormal. Normal working hours are considered to be 10:00-18:00, as this was determined to be the 8-hour interval with the most commits. In Figure 9, the outside office hours cluster is marked with black, the average cluster with blue, and the office hours cluster with green.

Table 11 takes a look at the paid/unpaid status of the developers based on manual data gathering, as explained in Section 3.3.1. We can see that volunteers are much more likely to work outside of office hours than paid developers. Of all volunteer developers, 47.9% belong to the outside office hours cluster, while only 10.1% of paid developers do. The majority of paid developers belong to the average cluster (50.2%), with another significant portion belonging to the office hours cluster (39.7%). The effect of background characteristics to office hour commits is shown on Table 12. No major differences with demographic information related to developers were found. The only statistically significant difference is between developers based in Europe and America, with the former committing slightly less during office hours than the latter.

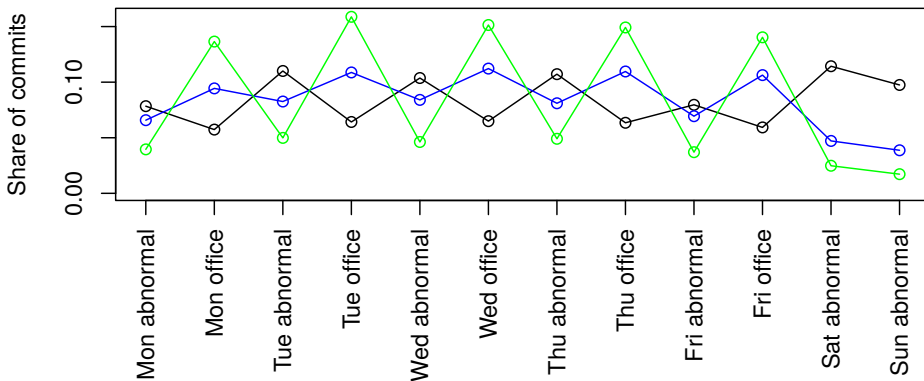


Fig. 9. Clusters of work patterns for the top 10% Firefox developers, with shares of commits per developer. (Reprinted, with permission, from Paper V. © 2018 IEEE).

Table 12. Median percentage of office hour commits by location, experience in the software industry, and position at Mozilla. (Reprinted, with permission, from Paper V. © 2018 IEEE).

Characteristic	# dev	# commits	office hours commits	Beginning of office hour period
Based in Europe	65	41,962	54.4%	9:45
Based in America	118	78,070	60.9%	10:00
Senior position	78	47,483	57.6%	10:00
Non senior position	197	130,468	59.5%	10:00
Manager position	19	7,957	63.1%	10:00
Non manager position	256	169,994	58.6%	9:30

The paper also examines commit size during office hours and outside office hours, but did not find significant differences in the size of the commits. Comparing wordclouds from commits messages reveals that reverts by developers themselves are more common outside office hours.

To sum it up, we found that the majority of software developers work during typical office hours, and that work outside office hours was more likely done unpaid developers. Otherwise, office hours were followed more rigorously in the local company, where all software developers were paid, compared to both Mozilla and Apache open source projects.

Author contribution: The author of this thesis was the third author of this paper, and responsible for the manual data gathering of paid and unpaid status of developers, communication with the local company from which data is used in the analysis, minor parts in data extraction, and writing the paper.

4.3.3 Paper VI: Chat activity is a better predictor than chat sentiment on software developers productivity.

Goal and motivation: The goal of the article is to explore the relationship between productivity of software developers and expressed chat sentiment. Productivity was measured as commits and lines of code changed. Expressed chat sentiment was measured with valence and arousal lexicons, as well as emoji and emoticons.

Results: The data is analyzed with pair-wise and step-wise regression models, both of them showing that the amount of chat messages sent, rather than the sentiment expressed in them, gives the best prediction ability on the amount of commits the developer will make during the same day.

Table 13. Pairwise linear regression models, where the productivity variable is predicted. The number provided is R^2 , and the sign signifies the sign of the coefficient. All predictors significant at $p < 0.05$ level. (Reprinted, with permission, from Paper VI. © 2020 Association for Computing Machinery).

Nr.	Variable	ncommitslog	nloclog
1	nchatlog	(+)0.330	(+)0.268
2	emoticon %	(+)0.038	(+)0.036
3	emot joy %	(+)0.117	(+)0.098
4	emot	(+)0.092	(+)0.062
	sadconfusionsurprise %		
5	sent high valence %	(+)0.034	(+)0.039
6	sent low valence %	(+)0.043	(+)0.042
7	sent min valence	(-)0.147	(-)0.108
8	sent max valence	(+)0.127	(+)0.103
9	sent high arousal %	(+)0.100	(+)0.100
10	sent low arousal %	(+)0.143	(+)0.130
11	sent minimum arousal	(-)0.196	(-)0.158
12	sent maximum arousal	(+)0.175	(+)0.146

Table 13 shows R^2 values for the pairwise linear regression models, where both productivity metrics, the number of commits, and the lines of code changed transformed to a logarithmic scale of ten, are predicted with various sentiment analysis variables. For both the number of commits and lines of code changed, the highest R^2 values are achieved by the logarithm number of chat messages. Other variables achieving R^2 value of over 0.14 are minimum valence, low arousal, minimum arousal, and maximum arousal for commits, and minimum and maximum arousal for lines of code changed. Scores related to arousal thus seem to perform marginally better than scores related to valence.

Table 14 shows the step-up build of a multivariate regression model predicting the number of commits. After weekday control variables, the first variable to be added (which lowers the AIC score the most) is the number of chat messages sent. After this is the variable related emojis and emoticons related to sadness, confusion and surprise. The remaining variables in order are maximum valence, low arousal, high valence, and lastly emoticon related to joy. Of note is the R^2 value achieved by the model: adding the number of chat messages increases it from 0.003 to 0.330, but the remaining five variables related to sentiment analysis only increase R^2 score 0.039 further.

The original publication also shows a step-up build of a multivariate regression model predicting the number of lines of code changed, with largely similar results.

Based on these results, it can be said that the number of chat messages sent had a larger predictive power than chat sentiment on productivity.

Table 14. Step-wise step-up build of multivariate regression model predicting the number of commits. P-value of the last added variable is shown. (Reprinted, with permission, from Paper VI. © 2020 Association for Computing Machinery).

Added variable	AIC	R^2	p-value
weekdays	5030.2	0.003	-
nchatlog	4396	0.330	< 2e-16
emot	4353.5	0.348	2.88e-11
sadconfusionsurprise %			
sent max valence	4328.1	0.360	1.75e-07
sent low arousal %	4315.5	0.365	0.000138
sent high valence %	4312.2	0.367	0.022
emot joy %	4310.5	0.369	0.054

Author contribution: The author of this thesis is the first author of the paper, and had the main role in analysis and writing of the paper.

4.3.4 *Main contributions and findings*

- Finding that the majority of software developers work during office hours, with more outside office hour work done by unpaid developers.
- Effect of location, seniority or job-title minimal to non-existent in observed data.
- Finding that the number of chat messages is a better predictor of productivity than sentiment analysis variables in instant messaging.

4.4 **RQ4: How does developer well-being link to software repository variables?**

4.4.1 *Introduction*

Three papers answer this research question. First, Paper VII introduces a daily questionnaire developed for experience sampling studies. Data gathered with the experience sampling methodology is then analyzed together with simple count variables from software repositories in Paper VIII. In Paper IX, the analysis is extended with additional sentiment analysis variables, as well as semi-structured interviews. A summary of the research methods used can be found in subsection 3.3.2.

4.4.2 Paper VII: Daily questionnaire to assess self-reported well-being during a software development project

Goal and motivation: The goal of this paper was to produce a questionnaire to be used in an ESM study, performed in a software development context. The questionnaire developed in this paper is the questionnaire used in the ESM study, and the results are analyzed in Papers VIII and IX. The questionnaire includes six different questions, which attempt to measure hurry, stress, sleeping problems, interruptions, possibility to make independent decisions related to work and ineffective software development. These single items are from various surveys and questionnaires: items related to independence and interruptions are from Karasek's Job Content Questionnaire (Karasek et al., 1998); the item measuring Hurry is from the Harris stress index (Harris, 1989); the item regarding stress is originally from the general health questionnaire "GHQ-12" (Goldberg & Blackwell, 1970) and refers to distress; and lastly the question concerning sleeping problems is from the Jenkins scale (Jenkins et al., 1988). The items were also slightly modified to fit a five-point scale.

The questionnaire was run at a local software company, and a total of 528 answers were received from eight software developers over a period of eight months.

Results: The paper presents the construction of the questionnaire based on prior work from the field of psychology, a description of a trial run of the questionnaire, and the rationale for including different questions in it. It also describes the way in which the questionnaire data was acquired for Papers VIII and IX.

Figure 10 shows a graph of five-day moving average of the answers to the questionnaire. Figure 11 shows a graph where each respondent's answers were normalized. As can be seen, independence is consistently higher than other measures, with a mean of 4.6. Means for other variables are less than 2.5.

The original paper also includes correlation analysis, where missing values were handled by carrying back observed values. This can be argued for by how the question is asked: "how frequently the condition had occurred since last time the questionnaire was answered". The answers were also transformed to time-series, and the trend and seasonal components for these time-series were removed. Even after this treatment, a positive statistically significant correlation was found between all questionnaire answers but Independence.

Gaussian graphical models and their contemporaneous network highlight strong auto-correlation of all questionnaire variables, meaning the best predictor for all questionnaire variables was the prior answer to the same question. The temporal network also shows

slight predictive power of sleeping problems for independence and ineffective software development.

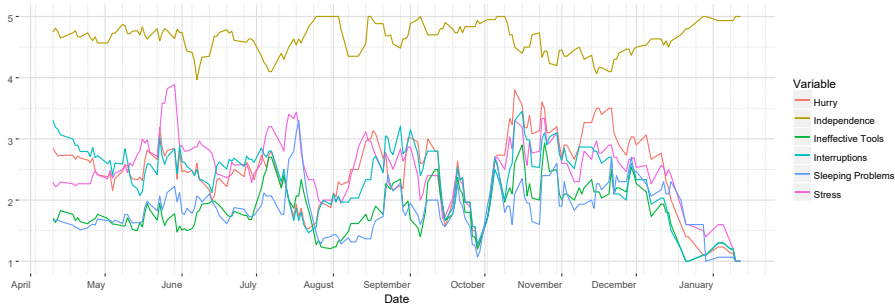


Fig. 10. Graph showing questionnaire answers with a five-day moving average. (Reprinted, with permission, from Paper VII. © 2018 Association for Computing Machinery).

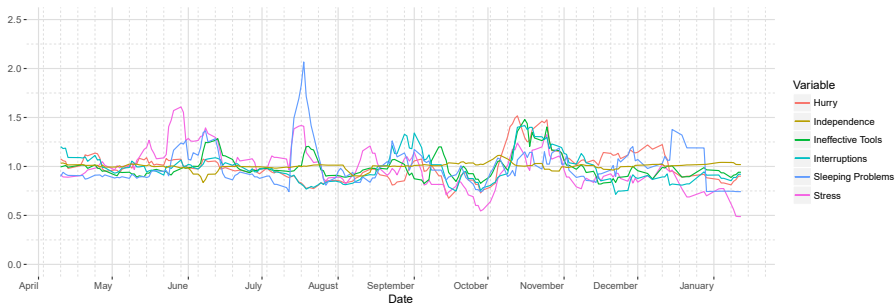


Fig. 11. Graph showing questionnaire answers, with the values normalized around zero. (Reprinted, with permission, from Paper VII. © 2018 Association for Computing Machinery).

Author contribution: The author of this doctoral dissertation had the main role in writing the paper and did the analysis presented in the paper.

4.4.3 Paper VIII: Using experience sampling to link software repositories with emotions and work well-being.

Goal and motivation: This study is to explore the possible links between affective states and simple software repository metrics. This was done by utilizing ESM methodology, together with mining software repository metrics. We ran a daily questionnaire constructed in Paper VII, for a period of eight months in a single software project, and acquired a total of 528 answers from eight respondents.

Results: Inter-coder agreement was calculated using the measure Krippendorff's α (Krippendorff, 2011), and the values produced are shown in Table 15. A value of 1 signifies perfect agreement, 0 signifies that units are statistically unrelated, and -1 means perfect disagreement. As can be seen in Table 15, all questionnaire variables show disagreement.

Table 16 shows a model where the number of lines of code changed is predicted with all questionnaire variables, both from the same day. For this model, both predictor and outcome variables are logistic (that is high or low), and control variables for weekly seasonality are included in the model. Four different predictors are statistically significant for the model, as they have a p-value of less than 0.05 in the last column. These are stress, sleeping problems, hurry and independence. The most significant of these is surprisingly hurry, which has a negative relationship with lines of code changed. The relationship can be found under the column Est., short for estimate, where a positive number shows a positive relationship and a negative number a negative relationship. Sleeping problems also have a negative relationship with lines of code changed; conversely, the relationship between lines of code changed between stress and independence is negative. The original publication also has a similar logistic regression model predicting the number of commits, which has similar statistically significant predictors, but this time it did not include independence.

Table 17 shows the logistic regression model, where the number of chat messages sent is predicted with the questionnaire variables. Again, for this model, both predictor and outcome variables are logistic. Statistically significant relationships between variables are shown in bold, where the p-value is less than 0.05. The sign of the estimate (Est.) tells the direction of the relationship. Sleeping problems have a negative relationship and independence a positive relationship with the number of chat messages sent. This means that developers reporting fewer sleeping problems and a higher ability to make independent decisions sent more chat messages. Here, model performance is not affected by lagged variables, as in Table 18.

The paper also shows logistic regression models, where a binary outcome variable derived from questionnaire answers is predicted with simple count variables derived from software repositories from the previous day, which are transformed to a logarithmic scale 10. Table 18 shows five of these models, where sleeping problems are predicted with previous days' repository variables. First, this is done separately with the number of commits, lines of code changed, and chat messages sent. After which, the best performing pair is shown, and lastly all three together. Control variables are added to the model as predictors, including dummy weekday variables, as well as previous days' answer for auto-correlation. These control variables and the high auto-correlation

Table 15. Inter-coder agreement of the respondents. (Reprinted, with permission, from Paper VIII. © 2018 Association for Computing Machinery).

Variable:	Krippendorff's α :
Hurry	-0.0993
Independence	-0.119
Ineffective Software Development	-0.178
Interruptions	-0.161
Sleeping Problems	-0.214
Stress	-0.155

account for the remarkably high accuracy of the models, with a 10-fold AUC between 0.863 and 0.873. For sleeping problems, all three repository variables are statistically significant predictors for the model when they are used alone. Only lines of code changed stays statistically significant when used in a pair of predictors, and none of them are statistically significant at below 0.05 p-value when all three are used as predictors together. These p-values are shown in the table in bold, implying that there are statistically significant relationships between these variables. Original publication includes five additional tables for each questionnaire variable, each with five models predicting the questionnaire variables in the same way.

Thus, several links between simple repository variables and questionnaire variables are found when logistic regressions and binning are used. Perhaps surprisingly, perceived hurry is negatively related to productivity metrics. Other links are more in line with what could be expected, *e.g.*, sleeping problems having negative relationship with both productivity and chat messages sent. In relation to paper IX, of note is the the binning of variables, adding of the control variables to the models, both of which have an impact on the results.

Author contribution: The author of the thesis is the first author, and was responsible for a major part in organizing the collaboration with the company, acquiring the data and its analysis, and lastly of writing of the paper itself.

4.4.4 Paper IX: Individual differences limit predicting well-being and productivity using software repositories: a longitudinal industrial study

Goal and motivation: This paper is an extension of Paper VIII. It extends the previously presented work in three significant ways: by adding multiple variables related to sentiment analysis, by using factor analysis to investigate relationships with extracted

Table 16. Logistic regression model for technical productivity (number of lines changed), controlled for weekly seasonality. Significant p-values (0.05) are shown in bold. (Reprinted, with permission, from Paper VIII. © 2018 Association for Computing Machinery).

Accuracy measures:

AIC: 581.77 AUC: 0.72 F1: 0.52 Prc: 0.71 R: 0.42

Variable	Est	Std. E	z value	Pr(> z)
stress	0.79	0.32	2.46	0.014
sleep	-0.74	0.27	-2.76	0.0058
hurry	-1.47	0.26	-5.72	1.065e-08
interruptions	0.094	0.31	0.3	0.76
ineffective	0.48	0.34	1.41	0.16
independence	0.56	0.22	2.50	0.012

Table 17. Logistic regression model for social interaction (number of chat messages), controlled for weekly seasonality. Significant p-values (0.05) are shown in bold. (Reprinted, with permission, from Paper VIII. © 2018 Association for Computing Machinery).

Accuracy measures

AIC: 520.67 AUC: 0.77 F1: 0.5 Prc: 0.73 R: 0.44

Variable	Est	Std. E	z value	Pr(> z)
stress	0.01	0.33	0.03	0.97
sleep	-0.61	0.26	-2.31	0.021
hurry	-0.21	0.26	-0.82	0.41
interruptions	-0.50	0.32	-1.56	0.12
ineffective	0.46	0.35	1.33	0.18
independence	1.9	0.27	7.03	2.093e-12

repository and sentiment analysis variables, by using a mixed effects models with autocorrelation structure to predict questionnaire answers and productivity, and lastly by using semi-structured interviews to further explain the quantitative results. The measurement model used is explained in Section 3.3.2 and Figure 4. In practice, this means combining multiple variables into a single factor with weights; for example, instead of using lines of code or number commits, these measures are combined into a single productivity factor.

Results: Table 19 presents linear mixed effects models predicting questionnaire answers with previous days' repository variables for all the respondents. The marginal

Table 18. Models for sleeping problems controlled for weekly seasonality and auto-correlation. Five different models are shown in the table columns. Significant p-values ($\alpha = 0.05$) are shown in bold. (Reprinted, with permission, from Paper VIII. © 2018 Association for Computing Machinery).

Model:	$\log(\text{ncommits})_{T-1}$	$\log(\text{nloc})_{T-1}$	$\log(\text{nchat})_{T-1}$	$\log(\text{nloc})_{T-1} + \log(\text{nchat})_{T-1}$	$\log(\text{ncommits})_{T-1} + \log(\text{nloc})_{T-1} + \log(\text{nchat})_{T-1}$
z-value	-2.345	-2.807	-2.249	-2.397 & -1.688	0.752 & -1.659 & -1.774
p-value	0.019	0.005	0.0245	0.0165 & 0.0915	0.4519 & 0.0971 & 0.0761
AIC	158.66	156.16	159.19	155.35	156.77
10-fold AUC	0.869	0.872	0.873	0.873	0.863
Precision	0.926	0.927	0.923	0.931	0.933
Recall	0.939	0.941	0.931	0.941	0.944
F1 Score	0.931	0.933	0.926	0.935	0.938

R^2 value represents the variance explained by the fixed effects (*i.e.*, repository variables), while the conditional R^2 value is interpreted as a variance explained by the entire model including fixed and random effects (*i.e.*, control variables such as weekdays, dates and respondent identification variable). The null model in Table 19 refers to a model without fixed effects.

Few statistically significant predictors can be found in this general model, including productivity having a positive relationship with hurry and stress, positive valence having a positive relationship with independence, and high arousal having a positive relationship with both interruptions and ineffective software development. However, when looking at the R^2 values, we can see that the marginal R^2 value differs between 0.01 and 0.02, while the conditional R^2 value is between 0.39 and 0.83. Thus, random effects account for the majority of the predictive ability of the model, and the individual identifier for the majority of the predictive ability of the random effects.

Data from individual developers were further investigated by making mixed effects model predicting questionnaire answers. Here only models for predicting hurry and sleeping problems in Tables 20 and 21. For conciseness sake, models predicting all the questionnaire answers for individual developers can be found in the original publication. Several statistically significant predictors can be found, *e.g.*, productivity for developer B when predicting hurry and high arousal for developer A when predicting sleeping

Table 19. Generalized linear mixed models predicting questionnaire variables with the previous working days' repository variables. A p-value of 0.05 or less is denoted in bold. (Under CC BY 4.0 license from Paper IX © 2021 Authors).

Predicted:	Hurry	Stress	Sleep	Inter- ruptions	Ineffe ctive	Indepe ndence
prod	0.07 (0.02)	0.19 (< 0.001)	0.06 (0.28)	-0.02 (0.71)	0.02 (0.47)	-0.03 (0.20)
nchat	-0.03 (0.34)	-0.06 (0.27)	0.05 (0.33)	-0.01 (0.89)	-0.02 (0.62)	0.01 (0.89)
pval	-0.03 (0.29)	0.04 (0.57)	0.08 (0.20)	-0.08 (0.14)	0.02 (0.67)	0.10 (< 0.001)
nval	0.03 (0.32)	-0.03 (0.58)	-0.09 (0.15)	0.03 (0.52)	0.04 (0.31)	0.04 (0.09)
har	0.02 (0.64)	0.02 (0.77)	0.02 (0.77)	0.13 (0.01)	0.09 (0.03)	-0.04 (0.11)
lar	-0.06 (0.10)	<-0.01 (0.93)	-0.04 (0.49)	-0.05 (0.23)	-0.01 (0.71)	-0.02 (0.48)
joyemo	0.01 (0.79)	0.03 (0.26)	-0.03 (0.42)	-0.02 (0.52)	-0.01 (0.53)	0.02 (0.22)
scsemo	-0.05 (0.05)	0.44 (0.25)	-0.20 (0.61)	0.28 (0.38)	0.31 (0.18)	0.35 (0.03)
failure event	-0.06 (0.07)	0.03 (0.56)	0.01 (0.91)	-0.04 (0.38)	<-0.01 (0.95)	-0.02 (0.42)
meeting	-0.13 (0.001)	0.01 (0.92)	<0.01 (0.97)	-0.01 (0.88)	0.04 (0.29)	-0.04 (0.13)
Random effects						
residual						
stddev :						
respondent	0.61	0.85	0.77	0.71	0.50	0.41
weekday	0.00	0.00	0.00	>0.01	0.00	>0.01
date	0.00	0.00	0.01	>0.01	0.05	>0.01
Marginal R^2	0.01	0.02	0.01	0.01	<0.01	0.01
Conditional R^2	0.76	0.39	0.44	0.67	0.83	0.68
Null Model C. R^2	0.74	0.39	0.44	0.67	0.80	0.66

Table 20. Generalized linear mixed models predicting hurry for the next day with today's repository variables. A p-value of 0.05 or less is denoted in bold. (Under CC BY 4.0 license from Paper IX © 2021 Authors).

Variable:	DevA	DevB	DevC	DevD
prod	0.11 (0.07)	-0.53 (< 0.001)	0.08 (0.47)	
nchat	-0.02 (0.80)	0.15 (0.23)	-0.09 (0.51)	
pval	-0.06 (0.56)	-0.51 (< 0.001)	-0.09 (0.64)	
nval	-0.06 (0.43)	-0.19 (0.24)	-0.11 (0.44)	
har	0.12 (0.25)	0.06 (0.61)	-0.28 (0.14)	
lar	-0.02 (0.74)	-0.04 (0.71)	-0.12 (0.41)	
joyemo	-0.06 (0.08)	0.02 (0.71)	0.20 (0.03)	
scsemo	0.37 (0.30)	-0.99 (0.18)	-0.08 (0.93)	
meeting	-0.03 (0.74)	-0.22 (0.24)	-0.02 (0.90)	
failure event	-0.11 (0.08)	-0.05 (0.68)	-0.08 (0.47)	
Marginal R^2	0.11	0.26	0.10	
Conditional R^2	0.11	0.26	0.10	
Null Model C. R^2	<0.01	0	0	

Table 21. Generalized linear mixed models predicting sleeping problems for the next day with today's repository variables. A p-value of 0.05 or less is denoted in bold. (Under CC BY 4.0 license from Paper IX © 2021 Authors).

Variable:	DevA	DevB	DevC	DevD
prod	0.19 (0.11)	-0.41 (0.07)	-0.03 (0.77)	-0.30 (0.57)
nchat	<-0.01 (0.99)	0.29 (0.13)	-0.19 (0.13)	0.05 (0.72)
pval	-0.06 (0.80)	-0.28 (0.11)	0.10 (0.55)	-0.04 (0.84)
nval	-0.23 (0.16)	-0.05 (0.82)	-0.14 (0.31)	-0.17 (0.35)
har	0.54 (0.02)	0.04 (0.83)	-0.16 (0.38)	0.07 (0.73)
lar	-0.03 (0.84)	-0.12 (0.48)	0.09 (0.51)	0.27 (0.19)
joyemo	-0.04 (0.63)	-0.05 (0.51)	-0.18 (0.04)	0.32 (0.44)
scsemo	0.30 (0.70)	-0.04 (0.97)	-0.13 (0.88)	-1.6 (0.75)
meeting	0.20 (0.20)	-0.49 (0.05)	0.01 (0.98)	0.20 (0.44)
failure event	-0.03 (0.84)	0.01 (0.95)	0.01 (0.91)	0.03 (0.77)
Marginal R^2	0.11	0.14	0.10	0.24
Conditional R^2	0.11	0.16	0.10	0.24
Null Model C. R^2	0	0	0	0

Table 22. Generalized linear mixed models predicting productivity during the same day. The general model and the four different individuals' models. A p-values 0.05 or less is denoted in bold. (Under CC BY 4.0 license from Paper IX © 2021 Authors).

Variable:	All	DevA	DevB	DevC	DevD
nchat	0.05 (0.35)	0.04 (0.73)	0.15 (0.15)	0.19 (0.19)	
pval	0.02 (0.74)	-0.02 (0.91)	-0.08 (0.42)	0.06 (0.78)	
nval	-0.05 (0.48)	-0.10 (0.51)	-0.05 (0.68)	-0.01 (0.94)	
har	-0.06 (0.39)	0.25 (0.16)	-0.15 (0.18)	-0.21 (0.30)	
lar	-0.07 (0.20)	0.06 (0.66)	0.03 (0.74)	-0.05 (0.76)	
joyemo	-0.02 (0.63)	-0.05 (0.42)	-0.02 (0.59)	0.04 (0.66)	
scsemo	-0.01 (0.97)	0.73 (0.49)	-0.50 (0.17)	-0.12 (0.91)	
meeting	-0.01 (0.96)	-0.01 (0.97)	0.15 (0.29)	0.04 (0.82)	
failure event	0.07 (0.18)	0.04 (0.69)	0.05 (0.63)	0.14 (0.25)	
stress	0.14 (0.01)	0.27 (<0.001)	-0.05 (0.74)	0.01 (0.93)	
sleep	-0.04 (0.48)	-0.10 (0.32)	-0.03 (0.75)	0.10 (0.53)	
hurry	-0.08 (0.25)	0.20 (0.42)	-0.30 (0.06)	0.01 (0.96)	
interruptions	0.01 (0.97)	-0.32 (0.29)	0.26 (0.06)	0.01 (0.92)	
ineffective	-0.04 (0.58)	0.59 (0.17)	-0.11 (0.39)	-0.04 (0.79)	
independence	-0.02 (0.81)	0.02 (0.95)	0.18 (0.31)	0.02 (0.99)	
Marginal R^2	0.03	0.19	0.20	0.05	
Conditional R^2	0.52	0.21	0.20	0.05	
Null Model C. R^2	0.49	0	0	0	

problems. However, it can be noted that the direction of the relationship (*i.e.*, the sign) between questionnaire variable and the predictors vary between the individuals.

It can be seen that the marginal R^2 values for individual developers vary between 0.10 and 0.26 for the presented models, and thus far surpass the general models R^2 value of 0.03. Hence, developers' well-being varied individually rather than in a collective manner. Developers' behaviour, or software engineering actions mined mainly from software repositories, were not good general predictors of well-being. Rather, it was individuals modeled as a random effect that the variable that explained differences in well-being.

Table 22 presents general and three individual mixed effects models predicting productivity during the same day. For the fourth individual, the model did not converge, which was a common problem for individual developers' models. Here, sentiment analysis variables and job event variables are used together with questionnaire answers to predict productivity variables derived from number of commits made, lines of

code changed, and number of files changed, as explained in Section 3.3.2. Only the statistically significant predictor for productivity in the general model is stress, which has a positive relationship with productivity. As can be seen from the R^2 values, the conditional R^2 value (0.52) accounts for the majority of the variance in the general model, with the marginal R^2 being just 0.03.

Data from individual developers were further investigated by making models with individual data. For statistically significant predictors, for Developer A, stress has a positive relationship with productivity. For individual developer models, the developers identifier is no longer part of the random effects, which solely consist of auto-correlation structure, date and weekday variables. Conditional R^2 values are close to the marginal R^2 values, meaning random effects do not add much to predictive ability. Thus it was found that models of well-being and productivity developed per individual performed better than general models. General model predicting productivity had a marginal R^2 value of 0.03, while in the individual models top marginal R^2 value was 0.26.

Interviews helped to contextualize the study. Based on interviews, no big deadline pressures or prolonged time pressures were felt during the project, which might in part make prediction of time pressure more difficult. Figure 11 shows a few spikes during July and October for sleeping problems, as well as all the questionnaire variables. In the interviews, contributing factors outside of work were mentioned. One contextual factor related to work here is a separate operations team responsible for complex issues in hosting the developed service. The selection of the software project investigated in the study probably influences the results significantly, but this is left for future work. There is an inherent problem in convincing a time-starved project to answer daily questionnaires.

In the interviews, one developer noted feeling hurried when not having the whole day for coding, but instead had to participate in meetings, quality assurance or job training-related events. Also, instant messaging was not reported to be interruptive when it was done in the main project-wide chat channel. However, one developer expressed feeling interrupted when getting private messages, as a response from them specifically was needed. The leadership style of the project manager was described to be facilitating and supportive, and the developers described the project as having plenty of independence in day-to-day operations.

Author contribution: The author of this doctoral thesis is the first author, and was responsible for the analysis presented in the paper and had the main role in the data gathering and writing of the paper.

4.4.5 Main contributions and findings

- Logistic regression models showing relationships between questionnaire variables and software repository variables.
- Statistically significant relationship between productivity and stress in the general linear mixed effects model.
- Importance of the individual shown in both the general model and the individual models: as a random effect, individuals explain most in the general model; in the individual models, different predictors have different predictive abilities for questionnaire variables.

5 Discussion

This chapter reports the main findings of the studies presented in Section 4 and answers the research questions presented in Section 1. The results are then discussed within the context of the research questions presented in this thesis. The chapter ends by presenting and discussing threats to construct, internal, external and conclusion validity. More comprehensive discussions can be found in the original publications regarding both results and threats to validity.

5.1 RQ1 - What is the current state of research related to sentiment analysis and time pressure in the current literature?

Answer: We identified 1270 articles in the case of time pressure and 6,996 articles in the case of sentiment analysis. This resulted in 79 topics for time pressure and 109 topics in total for sentiment analysis. Topic analysis and qualitative coding of studies showed how broadly these areas have been studied, *e.g.*, qualitative coding in the case of time pressure, found 12 topics related to different occupations, while in the case of sentiment analysis 41 topics were related to different application domains. Word-clouds provided more information on the contents of the topics.

Discussion: Both papers show the challenge of reviewing very broad topics in scientific literature that are too broad to be investigated with more traditional literature review methodologies. In the case of time pressure, reviewing the topic "software engineering" already showed many papers were related to quality assurance, which was later confirmed in Paper III.

A list of hypotheses was inferred from the general literature for time pressure for potential operationalizations in software engineering. When conducting primary studies, we did not study in detail any of these hypotheses. This is mostly because of our ability to acquire data for this, and thus our datasets had limited information on aspects of the software quality. One of the studies part of the review (Hussein, 2011) observed a significant rise in the emotions of participants under time pressure. In Paper IX, our sentiment analysis variables were not able to predict answers to hurry question in the questionnaire.

In Paper II, a nearly 50-fold increase in sentiment analysis studies between 2005 and 2015 is observed, making sentiment analysis one of the fastest growing research areas. The studies are also scattered in various publications, with the top 15 publication venues representing slightly less than one third of the papers in total. It was also noted that

papers prior to 2013 tended to focus more on product reviews, whereas more recent studies have focused more on social media posts.

5.2 RQ2 - What is the current state of research related to time pressure in software engineering?

Answer: Our selection criterion found 102 papers that focus on, or presented evidence related to, time pressure in software engineering context. Based on these sources, a list of metrics and operationalizations of time pressure in software engineering were summarized. A systematic map based on the stages of software development process with complementary categories found the biggest categories to be cost estimation, which is one cause of time pressure, and quality assurance, where the effects are most highly felt. Effects of time pressure on individuals were also noted, which included negative effects on individual well-being. A systematic review on the effects of time pressure on efficiency and quality found support for increased efficiency and lowered quality in most empirical studies. The knowledge produced by the review was summarized and presented in a practitioner-oriented magazine.

Discussion: The result of increased efficiency under time pressure from purely empirical studies is partly in contradiction with cost and process simulation models overviewed. The majority of cost and process simulation models support an overall increase of effort for a software project with a compressed schedule. However, it must be noted that this is not as straightforward for all found models. For some models, the compressed schedule also increases the error rate, and thus the increase in effort can come from rework needed for the software according to the model. We believe there are several contextual factors that can have an effect on these results in general. First, different time scales play a role, as experiments typically last hours and long projects usually take years. Efficiency gains can be pronounced in the short term, while exhaustion and other negative effects on individual well-being become pronounced in the longer term. Second, managerial decisions can play a role in how time pressure or compressed schedule is experienced and felt by individual developers: depending on the situation, overtime hours can have a more negative effect on individual well-being compared to fewer features or requirements on the quality of the software. Differentiating how individuals experience time pressure can be achieved with the Challenge-Hindrance framework (LePine et al., 2005).

There were multiple studies reporting that agile ways of working mediated the effects of time pressure on individuals. In a seminal psychological, experiments by Ariely and Wertenbroch (2002) highlights that performance increases with aggressive

intermediate deadline setting. Thus, we believe that intermediate deadlines are behind the mediating effects of agile methodologies, which balance required effort more evenly over longer periods of time. This should help to avoid bigger deadline crunches, which might be more common with a single deadline.

5.3 RQ3 - Can sentiment analysis and commit activity times explain developer productivity and behaviour?

Answer: Commit activity times can explain some developer behaviour. Studying timestamps of commits established that commits follow a pattern following a weekly rhythm, where fewer commits are made during the weekend. Similarly, we found that commits follow a circadian rhythm, where more commits are done during office hours and less in the evening and during the night. No significant differences in commit size between office hours and outside office hours were found. When looking closer at Mozilla projects, we find that paid developers work more during office hours and unpaid developers work more during outside office hours. Other background information had a limited effect on commit activity times, with the types of positions at the company not playing a role and the only statistically significant difference existing between developers from America and Europe in office hours commits.

The ability of sentiment analysis to explain developer productivity was very limited in our study. Investigating developer chat logs with sentiment analysis did not provide strong links between expressed sentiment and productivity variables related to code commits. Instead, the amount of chat messages sent achieved higher R^2 values predicting lines of code changed and number of commits made than sentiment analysis variables. Thus, the amount of chat messages was a better predictor than the type of sentiment expressed in the chat.

Discussion: The previous study by Wang et al. (2012) found that scientists seem to work more during the night than the software engineers investigated in Paper V. Based on scientific paper downloads, the number of downloads during weekends accounts for over 60% of the downloads during weekdays, meaning considerable amount of time is spent working during weekends. Working during the night for US- and Germany-based scientists are also more common than for developers in our study. Thus, overtime and outside office hours work seem to be more common in academia than software industry.

While our study failed to establish clear links between developer sentiment and productivity, other studies in the software engineering domain have had trouble with sentiment analysis in general. Both a study by Jongeling, Sarkar, Datta, and Serebrenik (2017) and by Lin et al. (2018) roughly point out that neither general purpose nor

software engineering-specific sentiment analysis tools agree with manual labeling or with the results of each other in software engineering context.

A very possible reason for why sentiment analysis failed to explain productivity might be that it captures the amount of teamwork. In the work of Cataldo, Wagstrom, Herbsleb, and Carley (2006), social congruence, that is, the fit between task dependencies and coordination activities, was found to shorten development time. Future work in using communication logs of instant messaging applications should investigate this possibility. Furthermore, based on the interviews done in paper IX, the discussions in the instant messages analyzed are work oriented. Thus, it might be reasonable to assert that information flow and the level of teamwork between developers is a better predictor of productivity than feeling and sentiment expressed in those messages. It must be, however, noted that private messages were not used in any of the analyses done in Papers VI and IX, in which questions about subjects where the respondent was seen as an expert were asked.

5.4 RQ4 - How does developer well-being link to software repository variables?

Answer: Our results from the longitudinal study presented in paper IX find that developer well-being varied individually rather than in a collective manner. In the general models, variables related to developer behaviour and sentiment are modeled as fixed effects and are not good predictors of self-reported well-being or productivity. However, the individual modeled as a random effect explains the differences in well-being and software repository variables. Further investigations into individual developers produced models showed that fixed effects, that is the predictors from the repositories, could predict developer well-being better. In practice, this meant that the measure of variance explained by fixed effects, marginal R^2 value, got as high as 0.26. In comparison, for the general model, the marginal R^2 was from less than 0.01 to 0.02. Another thing of note is the fact that variables related to instant messaging could not be linked to negative effects on well-being, which could be expected based on prior literature.

Discussion: However, when we used logistic regression models by binning both questionnaire and simple count repository variables, many links were found in paper VIII. Furthermore, if the individual modeled as random effect was removed from the generalized mixed effects models, paper IX's results would be more similar to paper VIII. Thus, perhaps a higher number of respondents could raise the marginal R^2 values, that is, the fixed effects of the models, and thus establish better links between repository variables and software developer well-being. This result is also quite similar to affective

computing literature, where the levels of stress and the individual being predicted have a significant effect on the prediction accuracy. Going from two levels of stress to three levels of stress significantly lowers prediction accuracy in the paper by Cho et al. (2017). Similarly, in the study predicting call-center worker stress by skin-conductance by Hernandez et al. (2011), the prediction accuracy of the within-person models is 78%, falling to 73% for the between-subjects general model.

It was also noted that we could not observe any clear negative links between variables related to instant messaging and developer well-being. Prior work, such as Sykes (2011) and Cameron and Webster (2005), have noted increased interruptions to work because of instant messaging. One of the interviewed developers noted the same, but only when using private messages instead of project-wide chat. This was because a response specifically from them was needed. Based on the interviews, we explain our results with how instant messaging was used, that is, between the project personnel more as a collaborative tool to coordinate expertise, rather than as a means for delivering commands or checking up on individual developers' progress. One contextual factor possibly affecting this was the leadership style, which was also described to be more facilitating at the company, which in turn allowed for more independence. These mediating effects negating negative effects would make sense per the Job Demands-Resources model.

The general model in Paper IX shows a positive relationship between stress and productivity. This could suggest further work related to affective computing and interventions, which were discussed in section 2.5. In practice this could mean using interventions to alleviate stress for software developers during times that would be deemed stressful from commit activity. As noted in previous literature for stress interventions, one big challenge would be using interventions in a way where the intervention does not become a stressor in itself. This task seems particularly daunting in a situation where a software developer is in the middle of a deadline crunch, and thus a sustainable pace of work is already out of the question for that particular deadline. One could assume that in such a situation, fast task completion would be the highest priority for the developer, and all attempts to lessen stress have high potential to increase it. Another problem related to interventions is the privacy aspect, where a developer might feel being monitored or singled out by interventions based solely on their own work activities, which are logged into the tools being used. Thus it might be a good idea to use it in a higher-level setting, meaning giving more anonymized information on the state of the project as a whole to the project manager, who might have better ability to mediate them through decisions such as more resources in the form of added time before deadline or additional developers.

5.5 Threats to validity

This section discusses the validity threats present in the original publications. The subsections are divided into areas of construct, internal, external, and conclusion validity. Thus, we follow the classification proposed by Wohlin et al. (2012).

5.5.1 Construct validity

Papers V, VI, VIII and IX deal in part with commits activity times and commit sizes. The timestamp itself only informs when the change was introduced to the version control system, not when the work towards that change was done. There is no obvious workaround to improve the accuracy. However, this is less of an issue when dealing with the local company's data, as the developers were instructed to do small commits more often.

The sentiment analysis performed in studies VI and IX is quite rudimentary, following from the fact that the development team used the Finnish language for instant messages. As the company did not want 3rd parties to have any access to the chat logs, this meant that instead of the chat logs, we translated the lexicons used in these studies. We also did not have full access to the data sources and couldn't investigate the accuracy of the analysis. Further improvement to sentiment analysis could probably be made by studying company-specific emoticon usage more in depth. Themes and topics outside work could also be discussed in the chatroom, though based on the interviews we believe this was the case in a clear minority of the time. Chat channels for these kinds of topics existed and were company wide instead of project wide, though the effects of these channels were left unstudied. Additionally, in the interviews, the developers themselves estimated that the topic was work-related in the clear majority of the cases.

The usage of single item measurements in experience sampling studies is a source of debate. The validity of "doubly concrete" constructs has been argued for by Rossiter (2002), meaning constructs for which the object and attribute of measurement are unambiguous and clear for the raters. Evidence supporting this view is also presented by a multitude of other studies, *e.g.* Bergkvist and Rossiter (2009) and Wanous, Reichers, and Hudy (1997). Both supporting and contradictory evidence, as well as discussion, can be found in an article by Fisher and To (2012). However, Fisher and To (2012) conclude that single item measurements are more valid when they are "straight forward unidimensional constructs in terms of current or very recent experience," rather than more complicated constructs that are rated retrospectively over a longer time span.

Thus, while the ratings were current, the broadly defined concept of ineffective software development included poor processes, performing tools, or poor communication with the development team. Based on the above paragraph, in my opinion, concepts other ineffective software development are more straightforward and should produce more valid data.

Factors outside of work have an effect on individual well-being, such as stress, and thus on the answers given to the questionnaire analyzed in Papers VII, VIII and IX.

5.5.2 Internal validity

One threat to internal validity concerning all secondary studies is publication bias, which has been defined as the occurrence of research in the published literature that is systemically unrepresentative of the population of completed studies by Rothstein, Sutton, and Borenstein (2005). For example studies with negative results might have a lesser chance of being published.

Another threat to validity concerning reviews is the selection of the databases with which literature is searched. For Papers I and II, we selected the Scopus database for the majority of the searches specifically because of the ease of exporting large amount of search results for further analysis. Both Google Scholar and Scopus were used for Paper III. There is an inherent trade-off in effort spent and the amount of results found when checking multiple databases for search results. We believe that performing backward snowballing helps in combating this bias, as the reference list of the sources does not depend on any individual search engine.

Selection of random effects when constructing generalized mixed effects models is a source of validity threats. Crawley (2002) supports the use of variables as fixed effects when there are not enough levels inside them. Furthermore, a thorough discussion by Bolker et al. (2020) sees the minimum level to be six. In our models, we used weekday variables and the respondent ID as random effects. Whether the weekday variable was fixed or random did not have a huge effect on the results. However, the respondent identifier does have a big effect on the models presented; results of Paper IX would be closer to Paper VIII without the respondent ID as a random effect.

There were model convergence issues in Paper IX. Model convergence is influenced by the complexity of random effects structures and sample size in generalized random effects structures (Barr, Levy, Scheepers, & Tily, 2013). Convergence issues were encountered when producing models for individual respondents, thus pointing at the limiting factor of sample size. Some issues were resolved by simplifying random effects

structure with different moving averages in the auto-correlation structure for individual developer models.

5.5.3 External validity

For Paper IX, we tried to contextualize our study by both describing the software project as best as we could and with the semi-structured interviews we performed. Describing the software project is a threat to the anonymity of the respondents for the study, which has a relatively low number of participants. The described contextual factors include company culture, which the project manager described as facilitating and allowing for independence for developers. The interviewed developers did not report any major time pressure in deadline crunches. Other factors of note were the agile way of working, pushing code to production daily without big integrations. Thus, we believe our results are replicable in this specific context.

5.5.4 Conclusion validity

Examining only a single project in Papers VI to IX lessens the generalizability of the results. Furthermore, in Paper IX the interviewees stated that no extraordinary pressures for delivery were felt during the project. Thus, detecting time pressure with repository variables might have been difficult because the developers' experience of it was very limited, and it might be easier in projects where more or even extreme time pressure is felt.

6 Conclusions

This chapter concludes this thesis by listing the contributions of the work to the larger body of knowledge. Implications for both academia and the industry are also discussed. Concluding with this section, the thesis reflects on the possibilities for future research and in extending the work.

6.1 Contributions

The following subsection lists the contributions of the studies and summarizes the conclusions in a few sentences. Contributions with a number in front of them are contributions where the author of this dissertation had a major role producing. The contributions listed are achieved by listing results from research questions of the original research papers, as well as the results sections of the original papers section of this thesis. Subsection academia discusses the results implications of these results for practitioners.

Academia

The bibliographic studies performed in papers I and II have the following contributions:

- Qualitatively coded topics from clustering for both 1) time pressure and sentiment analysis studies, highlighting how broadly these topics have been studied.
- 2) List of hypotheses inferred from the broader literature on time pressure that might help in detection of time pressure specifically in a software engineering context.
- Analysis of publication venues and citation analysis of sentiment analysis papers, showing a near 50-fold increase in the number of studies related to sentiment analysis from 2005 to 2015.
- 3) Summary of top cited papers related to sentiment analysis. These roughly include four categories of papers: reviews and overviews, online reviews, social media, and others.

The systematic review performed in Paper III, and of which results Paper IV summarized, has the following contributions:

- 4) A set of definitions of time pressure used in software engineering literature, which are shown in Table 1.

- 5) Tying the definitions to the theories of Yerkes Dodson’s law and Challenge and Hindrance time pressure. This is further explained in Section 2.1 and highlighted in Figure 7.
- 6) Analysis of publication venues and years of the selected literature on time pressure in software engineering. This is shown in the original publication.
- 7) Summary and categorization of reported causes of time pressure to effort estimation, project management and company culture. This is further discussed in Section 4.2.2 and the original publication.
- 8) Providing an overview on the studies of the effects of time pressure on individuals, which included effects on developer well-being. This is further discussed in Section 4.2.2 and the original publication.
- 9) Mapping the selected papers to different stages of software development processes and approaches. This is shown in Table 9, where the highest number of papers are related to Quality Assurance.
- 10) Systematic review of empirical studies on the effects of time pressure on software development efficiency and software quality, where the majority of studies support increased efficiency and reduced quality under time pressure. This contribution is shown in Table 10.

The repository mining studies performed in Papers V and VI have the following contributions:

- Overview of commit activity times for 86 open source projects, showing that commit times follow work weeks and the circadian rhythm. These are partly shown in Figure 8.
- 11) Investigation into paid/unpaid status of developers, as well as other background information of software developers, and their relation to commit information. Unpaid developers are more likely to commit outside office hours; the effect of other background information is very limited. These are shown in Tables 11 and 12.
- 12) Result that the number of chat messages sent was a better predictor of productivity than sentiment analysis variables in instant messaging. This is shown in Table 14.

The three papers reporting a field study in papers VII, VIII, and IX have the following contributions:

- 13) A questionnaire used to study well-being in software engineering context. It is introduced in Section 3.3.2 and paper VII.
- 14) Inter-rater agreement calculation into questionnaire variables shown in Table 15, which shows individual rating of all variables related to well-being.

- 15) Logistic regression models with binning showing relationships between software repository variables and developer well-being, examples of which are shown in Tables 16, 17, and 18. Models having 10-fold areas under the curve from 0.71 to 0.9 depending on the control variables used.
- 16) Generalized linear mixed models showing the importance of the individual in predicting software developer well-being and productivity. Statistically significant relationship between stress and productivity in the general model shown in Table 19.
- 17) Thus a demonstration of differing results based on the methodology used to analyze the data: differing results between logistic regression with binning and generalized linear mixed models.
- 18) Explorations into models made with individual's data, showing the variance in achieved R^2 values and the difference between predictors between individuals. Individual models are shown in Tables 20 and 21.

Industry

Most of the knowledge produced that is relevant for industry is from Paper III, and the industry summary of it is provided in Paper IV. This is because the paper deals with time pressure and scheduling-related issues in a very broad sense. The causes of time pressure were categorized roughly into three categories: effort underestimation, poor project planning and management, and lastly, company culture. However, it can be said that commercial pressures underlie all of these causes. While effort estimation and project management are their own areas of research in scientific literature, our literature search found descriptions of work environments where pressure and overtime work was the goal of the management.

The reviewed evidence strongly points towards increased efficiency and reduced quality under time pressure. According to the challenge-hindrane framework and real-life examples, development teams can have exceptional performance under time pressure, by making them focus on the most essential features. The negative effects of worse software quality and weakened developer well-being can become an issue in the longer term, especially with prolonged time pressure. A good example of worse software quality and increased rework in the reviewed literature was a workaround patch, which was implemented at least 12 times instead of a single fix (Lavallée & Robillard, 2015).

Other takeaways from the reviewed studies include tying a part of the success of agile methodologies to intermediate deadline setting. Based on the broader literature (Ariely & Wertenbroch, 2002), recommending the setting of intermediate deadlines can

be recommended even when not using Agile. Additionally, there was some evidence that the level of developer experience and the complexity of the development task play a role on the performance under time pressure. In general, it can be said that more experienced developers do better in less complex tasks under time pressure.

Takeaways from the primary study branch are more limited for industry, but are related to developer well-being and instant messaging. We did not find variables that would predict well-being for developers in general with software repository variables. However, some predictive ability could be found from the repository variables, but how they predicted well-being related variables varied from developer to developer. Usage of instant messaging was not tied to negative effects such as interruptions or increased stress, and based on this single project, usage of instant messaging can be recommended, provided it is not used for urgent matters, constant checking on workers or giving out orders, but rather coordination of expertise.

6.1.1 Future research

Future research related to timing and scheduling related issues in software engineering, as well as software developer well-being has multiple avenues to progress. Studies looking at time management and company culture could try to further quantify the predominance of pressured work environments in the software industry. Different opinions on priorities are voiced in the media by workers in the IT sector in Finland, such as a return to 70-hour work week and abandonment of long holidays (Leskinen, 2021). The work schedule of 996, working from 9am to 9pm six days per week, has also been under investigation in the Chinese IT sector (Zhang et al., 2020). Thus, while empirical evidence can explain the effects of different work cultures and priorities on health and well-being, people have different goals and values for their lives in general. Thus, to provide insight into cultural differences, surveys of software developers could investigate work cultures and values related to in different countries. Related to current events, investigating the effects of Covid-19 on software development work could provide meaningful information, such as the impact of lock-downs on the amount of remote work and the changes to the work-life balance of software developers.

B. A. Kitchenham (1992), Paper III, and Chapter 4.2.2 of this thesis note some different results between empirical studies and assumptions of software cost models on the effects of time pressure on efficiency and quality. We suspect that these differing results are in part explained by different time scales, different management decisions, and differing starting points in projects. Time scales here refer to the fact that experiments in a lab setting can take few hours for participants, while software projects typically take

months or years for software developers. We also suspect that how time pressure is actualized in the software project plays a role, as having less reusable code might be different than having to work overtime during a weekend from the Challenge-Hindrance framework point of view. The starting point in a project can also be vastly different based on how the previous project went, especially if the project personnel stay the same for the next project. Perhaps more efforts could be undertaken, which take these contextual factors into account and could better explain these differences in studies thus far published.

Future studies using experience sampling and repository mining to study work well-being in software development context can be improved. First of all, a higher sample size would benefit the analysis, and thus a higher number of respondents should be used if possible. To convince larger groups to respond to daily questionnaires over a period of time, perhaps the duration of the questionnaire could be shorter. This could mean a month or two instead of eight with an increased number of developers. A higher sample size would also make it possible to study the effects of individual differences on well-being better, such as possible differences between genders and the experience level of the developer. With enough respondents studying the effect of personality types (Eysenck, Barrett, & Saklofske, 2020), would be interesting. Replicating our experience sampling study in a different context and a larger sample size could also create evidence on the effects of contextual factors, such as leadership style and application of different software processes, tools, and methodologies.

Experience sampling could also be used to study other factors in software engineering. These could be the effects of different kinds of approaches, tools, techniques and ways of working affect the experiences of software developers. These factors could include the adoption of agile, resistance to organizational changes, and organizational justice in general. Relating to my research, perhaps having more items in the questionnaire for each factor could have yielded different results. I would personally opt for this if I were to try to reproduce the results of the field studies from Paper VII to IX.

During the research for this thesis, there was a considerable amount of effort spent into looking for ways to deal with time dependencies in the data. For example, all questionnaire variables showed considerable auto-correlation; that is, the questionnaire answers were significantly correlated with the questionnaire answers the day before. Thus, overviews on how these issues related to time and dependencies it can create in the data might be a worthwhile effort, specifically in software engineering and repository mining research. Investigations into how they are currently taken into account might also be justifiable.

References

- Adams, W. C. (2015). Conducting semi-structured interviews. *Handbook of Practical Program Evaluation*, 492–505.
- Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotogu Akaike* (pp. 199–213). Springer.
- Alliger, G. M., & Williams, K. J. (1993). Using signal-contingent experience sampling methodology to study work in the field: A discussion and illustration examining task perceptions and mood. *Personnel Psychology*, 46(3), 525–549.
- Andrés, A. (2009). *Measuring academic research: How to undertake a bibliometric study*. Elsevier.
- Ariely, D., & Wertenbroch, K. (2002). Procrastination, deadlines, and performance: Self-control by precommitment. *Psychological science*, 13(3), 219–224.
- Bakker, A. B., & Demerouti, E. (2007). The job demands-resources model: State of the art. *Journal of Managerial Psychology*, 22(3), 309–328.
- Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3), 255–278.
- Barrett, L. F. (1998). Discrete emotions or dimensions? the role of valence focus and arousal focus. *Cognition & Emotion*, 12(4), 579–599.
- Barton, K. (2009). *MuMin: multi-model inference*. Retrieved 2021-10-26, from <http://r-forge.r-project.org/projects/mumin/>
- Basten, D. (2017). The role of time pressure in software projects: A literature review and research agenda. In *eproceedings of the 12th international research workshop on information technology project management (IRWITPM)* (pp. 1–15).
- Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 436–442).
- Bergkvist, L., & Rossiter, J. R. (2009). Tailor-made single-item measures of doubly concrete constructs. *International Journal of Advertising*, 28(4), 607–621.
- Bhushan, N., Mohnert, F., Sloot, D., Jans, L., Albers, C., & Steg, L. (2019). Using a Gaussian graphical model to explore relationships between items and variables in environmental psychology research. *Frontiers in psychology*, 10, 1050.
- Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., . . . Steece, B. (2000). Cost estimation with COCOMO II. *ed: Upper Saddle River, NJ: Prentice-Hall*.

- Boehm, B. W. (1981). *Software engineering economics* (Vol. 197). Prentice-Hall Englewood Cliffs (NJ).
- Bolker, B., et al. (2020). *GLMM FAQ*. Retrieved 2021-01-18, from <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#should-i-treat-factor-xxx-as-fixed-or-random>
- Bollen, K. (2001). Indicator: Methodology. In N. J. Smelser & P. B. Baltes (Eds.), *International encyclopedia of the social & behavioral sciences* (p. 7282 - 7287). Oxford: Pergamon. Retrieved from <http://www.sciencedirect.com/science/article/pii/B0080430767007099> doi: <https://doi.org/10.1016/B0-08-043076-7/00709-9>
- Bozhkov, L., Georgieva, P., Santos, I., Pereira, A., & Silva, C. (2015). EEG-based subject independent affective computing models. *Procedia Computer Science*, 53, 375–382.
- Bradford, L. (2018, June). *Why we need to talk about burnout in the tech industry*. Retrieved 2021-10-26, from <https://www.forbes.com/sites/laurencebradford/2018/06/19/why-we-need-to-talk-about-burnout-in-the-tech-industry/> ([Online; posted 19-June-2018])
- Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7), 1145–1159.
- Brooks Jr., F. P. (1995). *The mythical man-month: Essays on software engineering, anniversary edition, 2nd ed.* Pearson Education India.
- Calefato, F., Lanubile, F., Maiorano, F., & Novielli, N. (2018). Sentiment polarity detection for software development. *Empirical Software Engineering*, 23(3), 1352–1382.
- Cameron, A. F., & Webster, J. (2005). Unintended consequences of emerging communication technologies: Instant messaging in the workplace. *Computers in Human behavior*, 21(1), 85–103.
- Cataldo, M. (2010). Sources of errors in distributed development projects: Implications for collaborative tools. In *Proceedings of the 2010 ACM conference on computer-supported cooperative work* (pp. 281–290).
- Cataldo, M., Wagstrom, P. A., Herbsleb, J. D., & Carley, K. M. (2006). Identification of coordination requirements: Implications for the design of collaboration and awareness tools. In *Proceedings of the 2006 20th anniversary conference on computer supported cooperative work* (pp. 353–362).
- Chittaro, L., & Sioni, R. (2014). Affective computing vs. affective placebo: Study of a

- biofeedback-controlled game for relaxation training. *International Journal of Human-Computer Studies*, 72(8-9), 663–673.
- Cho, Y., Bianchi-Berthouze, N., & Julier, S. J. (2017). DeepBreath: Deep learning of breathing patterns for automatic stress recognition using low-cost thermal imaging in unconstrained settings. In *2017 seventh international conference on affective computing and intelligent interaction (ACII)* (pp. 456–463).
- Chong, D. S., Van Eerde, W., Chai, K. H., & Rutte, C. G. (2011). A double-edged sword: The effects of challenge and hindrance time pressure on new product development teams. *IEEE Transactions on Engineering Management*, 58(1), 71–86.
- Cooper, C. L., Dewe, P. J., & O’Driscoll, M. P. (2001). *Organizational stress: A review and critique of theory, research, and applications*. Sage.
- Costa, J., Guimbretière, F., Jung, M. F., & Choudhury, T. (2019). Boostmeup: Improving cognitive performance in the moment by unobtrusively regulating emotions with a smartwatch. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2), 1–23.
- Costello, S. H. (1984). Software engineering under deadline pressure. *ACM SIGSOFT Software Engineering Notes*, 9(5), 15–19.
- Courage, C., & Baxter, K. (2005). *Understanding your users: A practical guide to user requirements methods, tools, and techniques*. Gulf Professional Publishing.
- Crawley, M. J. (2002). *Statistical computing: an introduction to data analysis using S-Plus* (No. 001.6424 C73).
- de F. Farias, M. A., Novais, R., Júnior, M. C., da Silva Carvalho, L. P., Mendonça, M., & Spínola, R. O. (2016). A systematic mapping study on mining software repositories. In *Proceedings of the 31st annual ACM symposium on applied computing* (pp. 1472–1479).
- Demerouti, E., Bakker, A. B., Nachreiner, F., & Schaufeli, W. B. (2001). The job demands-resources model of burnout. *Journal of Applied psychology*, 86(3), 499.
- Diener, E., Suh, E. M., Lucas, R. E., & Smith, H. L. (1999). Subjective well-being: Three decades of progress. *Psychological bulletin*, 125(2), 276.
- Droba, D. (1931). Methods used for measuring public opinion. *American Journal of Sociology*, 37(3), 410–423.
- Dueñas, S., Cosentino, V., Robles, G., & Gonzalez-Barahona, J. M. (2018). Perceval: Software project data at your will. In *Proceedings of the 40th international conference on software engineering: Companion proceedings* (pp. 1–4).
- Ebert, C., & Jones, C. (2009). Embedded software: Facts, figures, and future. *Computer*, 42(4).

- Ekkekakis, P. (2012). Affect, mood, and emotion. *Measurement in sport and exercise psychology*, 321.
- Ekman, P. (1999). Basic emotions. *Handbook of cognition and emotion*, 98(45-60), 16.
- Elovainio, M., Heponiemi, T., Jokela, M., Hakulinen, C., Penseau, J., Aalto, A.-M., & Kivimäki, M. (2015). Stressful work environment and wellbeing: What comes first? *Journal of occupational health psychology*, 20(3), 289.
- Epskamp, S., Borsboom, D., & Fried, E. I. (2018). Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, 50(1), 195–212.
- Epskamp, S., Waldorp, L. J., Mötus, R., & Borsboom, D. (2018). The Gaussian graphical model in cross-sectional and time-series data. *Multivariate behavioral research*, 53(4), 453–480.
- Eysenck, S. B., Barrett, P. T., & Saklofske, D. H. (2020). The Junior Eysenck personality questionnaire. *Personality and Individual Differences*, 109974.
- Faraway, J. J. (2016). *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press.
- Fellows, I. (2012). wordcloud: Word clouds. *R package version*, 2, 109.
- Fisher, C. D., & To, M. L. (2012). Using experience sampling methodology in organizational behavior. *Journal of Organizational Behavior*, 33(7), 865–877.
- Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., . . . Yang, J. (2005). Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1), 119–146.
- Fredrickson, B. L. (2000). Cultivating positive emotions to optimize health and well-being. *Prevention & treatment*, 3(1), 1a.
- Fucci, D., Scanniello, G., Romano, S., & Juristo, N. (2018). Need for sleep: the impact of a night of sleep deprivation on novice developers' performance. *IEEE Transactions on Software Engineering*, 46(1), 1–19.
- Gaudin, S. (2015, August). *Silicon valley's 'pressure cooker:' thrive or get out*. Retrieved 2021-10-26, from <https://www.computerworld.com/article/2972723/it-careers/silicon-valleys-pressure-cooker-thrive-or-get-out.html> ([Online; posted 18-August-2015])
- Gilb, T., & Finzi, S. (1988). *Principles of software engineering management* (Vol. 11). Addison-wesley Reading, MA.
- Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational researcher*, 5(10), 3–8.
- Goldberg, D. P., & Blackwell, B. (1970). Psychiatric illness in general practice: a detailed study using a new method of case identification. *Br med J*, 2(5707),

439–443.

- Graziotin, D., Wang, X., & Abrahamsson, P. (2015). Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering. In *Proceedings of the 7th international workshop on social software engineering* (pp. 25–32).
- Greene, S., Thapliyal, H., & Caban-Holt, A. (2016). A survey of affective computing for stress detection: Evaluating technologies in stress detection for better health. *IEEE Consumer Electronics Magazine*, 5(4), 44–56.
- Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1), 5228–5235.
- Harris, P. E. (1989). The nurse stress index. *Work & Stress*, 3(4), 335–346.
- Hassan, A. E. (2008). The road ahead for mining software repositories. In *2008 frontiers of software maintenance* (pp. 48–57).
- Healey, J. A., & Picard, R. W. (2005). Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on intelligent transportation systems*, 6(2), 156–166.
- Heitz, R. P. (2014). The speed-accuracy tradeoff: History, physiology, methodology, and behavior. *Frontiers in Neuroscience*, 8, 150.
- Hernandez, J., Morris, R. R., & Picard, R. W. (2011). Call center stress recognition with person-specific models. In *International conference on affective computing and intelligent interaction* (pp. 125–134).
- Hernandez, J., Paredes, P., Roseway, A., & Czerwinski, M. (2014). Under pressure: sensing stress of computer users. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 51–60).
- Hilbe, J. M. (2009). *Logistic regression models*. Chapman and hall/CRC.
- Hox, J. J., & Boeije, H. R. (2005). Data collection, primary versus secondary. *Encyclopedia of social measurement, Volume 1*.
- Hudson, S., Fogarty, J., Atkeson, C., Avrahami, D., Forlizzi, J., Kiesler, S., . . . Yang, J. (2003). Predicting human interruptibility with sensors: a wizard of oz feasibility study. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 257–264).
- Hussein, B. A. (2011). Quasi-experimental method to identify the impact of ambiguity and urgency on project participants in the early project phase. In *Proceedings of the 6th IEEE international conference on intelligent data acquisition and advanced computing systems* (Vol. 2, pp. 892–897).
- Hwang, M. I. (1994). Decision making under time pressure: A model for information systems research. *Information & Management*, 27(4), 197–203.

- Iacobucci, D., Posavac, S. S., Kardes, F. R., Schneider, M. J., & Popovich, D. L. (2015a). The median split: Robust, refined, and revived. *Journal of Consumer Psychology*, 25(4), 690–704.
- Iacobucci, D., Posavac, S. S., Kardes, F. R., Schneider, M. J., & Popovich, D. L. (2015b). Toward a more nuanced understanding of the statistical properties of a median split. *Journal of Consumer Psychology*, 25(4), 652–665.
- Iqbal, S. T., Zheng, X. S., & Bailey, B. P. (2004). Task-evoked pupillary response to mental workload in human-computer interaction. In *Chi'04 extended abstracts on human factors in computing systems* (pp. 1477–1480).
- Jenkins, C. D., Stanton, B.-A., Niemcryk, S. J., & Rose, R. M. (1988). A scale for the estimation of sleep problems in clinical research. *Journal of clinical epidemiology*, 41(4), 313–321.
- Jongeling, R., Datta, S., & Serebrenik, A. (2015). Choosing your weapons: On sentiment analysis tools for software engineering research. In *2015 IEEE international conference on software maintenance and evolution (ICSME)* (pp. 531–535).
- Jongeling, R., Sarkar, P., Datta, S., & Serebrenik, A. (2017). On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 22(5), 2543–2584.
- Juristo, N., & Vegas, S. (2011). The role of non-exact replications in software engineering experiments. *Empirical Software Engineering*, 16(3), 295–324.
- Kagdi, H., Collard, M. L., & Maletic, J. I. (2007). A survey and taxonomy of approaches for mining software repositories in the context of software evolution. *Journal of software maintenance and evolution: Research and practice*, 19(2), 77–131.
- Karasek, R., Brisson, C., Kawakami, N., Houtman, I., Bongers, P., & Amick, B. (1998). The Job Content Questionnaire (JCQ): an instrument for internationally comparative assessments of psychosocial job characteristics. *Journal of occupational health psychology*, 3(4), 322.
- Karasek, R., & Theorell, T. (1990). *Healthy work: Stress, productivity and the reconstruction of working life*. Basic Books.
- Kelly, J. R., & McGrath, J. E. (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.
- King, J. E. (2008). Binary logistic regression. *Best practices in quantitative methods*, 358–384.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.
- Kitchenham, B. A. (1992). Empirical studies of assumptions that underlie software

- cost-estimation models. *Information and software Technology*, 34(4), 211–218.
- Krippendorff, K. (2011). *Computing Krippendorff's alpha-reliability*. Retrieved 2021-10-26, from https://repository.upenn.edu/asc_papers/43/
- Kuutila, M., Mäntylä, M., Farooq, U., & Claes, M. (2020). Time pressure in software engineering: A systematic review. *Information and Software Technology*, 121, 106257.
- Larson, R., & Csikszentmihalyi, M. (2014). The experience sampling method. In *Flow and the foundations of positive psychology* (pp. 21–34). Springer.
- Lavallée, M., & Robillard, P. N. (2015). Why good developers write bad code: An observational case study of the impacts of organizational factors on software quality. In *2015 IEEE/ACM 37th IEEE international conference on software engineering* (Vol. 1, pp. 677–687).
- Lenberg, P., Feldt, R., & Wallgren, L. G. (2015). Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and software*, 107, 15–37.
- LePine, J. A., Podsakoff, N. P., & LePine, M. A. (2005). A meta-analytic test of the challenge stressor–hindrance stressor framework: An explanation for inconsistent relationships among stressors and performance. *Academy of Management Journal*, 48(5), 764–775.
- Leskinen, J. (2021, June). *Huawein suomalaisjohtajan radikaali ajatus: Seitsenpäiväinen työviikko heti käyttöön suomessa – ”minulla ei ole neljään vuoteen ollut kesälomaa”*. Retrieved 2021-10-26, from <https://www.kauppalehti.fi/uutiset/huawein-suomalaisjohtajan-radikaali-ajatus-seitsenpaivainen-tyoviikko-heti-kayttoon-suomessa-minulla-ei-ole-neljaan-vuoteen-ollut-kesalomaa/> b252553f-069d-4275-8cc6-c34e7f43b47f ([Online; posted 6-June-2021])
- Li, H., & Yamanishi, K. (2003). Topic analysis using a finite mixture model. *Information processing & management*, 39(4), 521–541.
- Lin, B., Zampetti, F., Bavota, G., Di Penta, M., Lanza, M., & Oliveto, R. (2018). Sentiment analysis for software engineering: How far can we go? In *Proceedings of the 40th international conference on software engineering* (pp. 94–104).
- Liu, B. (2009). Handbook chapter: Sentiment analysis and subjectivity. handbook of natural language processing. *Handbook of Natural Language Processing*. Marcel Dekker, Inc. New York, NY, USA.
- Lohan, G., Acton, T., & Conboy, K. (2013). The impact of group cohesiveness on decision-making outcomes under conditions of challenge and hindrance time pressure. In *8th pre-ICIS international research workshop on information*

- technology project management (IRWITPM 2013)* (p. 159).
- MacLean, D., Roseway, A., & Czerwinski, M. (2013). MoodWings: a wearable biofeedback device for real-time stress intervention. In *Proceedings of the 6th international conference on pervasive technologies related to assistive environments* (pp. 1–8).
- Majaranta, P., & Bulling, A. (2014). Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing* (pp. 39–65). Springer.
- Mäntylä, M. V., Graziotin, D., & Kuutila, M. (2018). The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Computer Science Review*, 27, 16–32.
- Mäntylä, M. V., Novielli, N., Lanubile, F., Claes, M., & Kuutila, M. (2017). Bootstrapping a lexicon for emotional arousal in software engineering. In *2017 IEEE/ACM 14th international conference on mining software repositories (MSR)* (pp. 198–202).
- Mäntylä, M. V., Petersen, K., Lehtinen, T. O., & Lassenius, C. (2014). Time pressure: A controlled experiment of test case development and requirements review. In *Proceedings of the 36th international conference on software engineering* (pp. 83–94).
- Mark, G., Iqbal, S. T., Czerwinski, M., Johns, P., Sano, A., & Lutchyn, Y. (2016). Email duration, batching and self-interruption: Patterns of email use on productivity and stress. In *Proceedings of the 2016 chi conference on human factors in computing systems* (pp. 1717–1728).
- Maruping, L. M., Venkatesh, V., Thatcher, S. M., & Patel, P. C. (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.
- Maule, A. J., Hockey, G. R. J., & Bdzola, L. (2000). Effects of time-pressure on decision-making under uncertainty: Changes in affective state and information processing strategy. *Acta Psychologica*, 104(3), 283–301.
- Maxwell, S. E., & Delaney, H. D. (1993). Bivariate median splits and spurious statistical significance. *Psychological bulletin*, 113(1), 181.
- McConnell, S. (1996). *Rapid development: taming wild software schedules*. Pearson Education.
- McDuff, D., Gontarek, S., & Picard, R. (2014). Remote measurement of cognitive stress via heart rate variability. In *2014 36th annual international conference of the IEEE engineering in medicine and biology society* (pp. 2957–2960).
- Merriam-Webster-Dictionary. (n.d.). *Sentiment*. Retrieved 2021-10-26, from

- <https://www.merriam-webster.com/dictionary/sentiment>
- Michie, S., Van Stralen, M. M., & West, R. (2011). The behaviour change wheel: a new method for characterising and designing behaviour change interventions. *Implementation science*, 6(1), 1–12.
- Miller, H. J., Thebault-Spieker, J., Chang, S., Johnson, I., Terveen, L., & Hecht, B. (2016). “Blissfully happy” or “ready to fight”: Varying interpretations of emoji. In *Tenth international AAAI conference on web and social media*.
- Molokken, K., & Jorgensen, M. (2003). A review of software surveys on software effort estimation. In *Empirical software engineering, 2003. ISESE 2003. proceedings. 2003 international symposium on* (pp. 223–230).
- MSR. (2021, March). *Mining software repositories*. Retrieved 2021-10-26, from <https://www.msrconf.org/> ([Online])
- Mullainathan, S., & Shafir, E. (2013). *Scarcity: Why having too little means so much*. Macmillan.
- Nakagawa, S., & Schielzeth, H. (2013). A general and simple method for obtaining R² from generalized linear mixed-effects models. *Methods in ecology and evolution*, 4(2), 133–142.
- Nan, N., & 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.
- Noroozi, F., Kaminska, D., Corneanu, C., Sapinski, T., Escalera, S., & Anbarjafari, G. (2018). Survey on emotional body gesture recognition. *IEEE transactions on affective computing*.
- Novielli, N., Calefato, F., Dongiovanni, D., Girardi, D., & Lanubile, F. (2020). Can we use se-specific sentiment analysis tools in a cross-platform setting? In *Proceedings of the 17th international conference on mining software repositories (MSR)* (pp. 158–168).
- Novielli, N., Calefato, F., Lanubile, F., & Serebrenik, A. (2021). Assessment of off-the-shelf se-specific sentiment analysis tools: An extended replication study. *Empirical Software Engineering*, 26(4), 1–29.
- Novielli, N., Girardi, D., & Lanubile, F. (2018). A benchmark study on sentiment analysis for software engineering research. In *2018 IEEE/ACM 15th international conference on mining software repositories (MSR)* (pp. 364–375).
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1–135.
- Paredes, P., & Chan, M. (2011). CalmMeNow: exploratory research and design of stress mitigating mobile interventions. In *Chi'11 extended abstracts on human*

- factors in computing systems* (pp. 1699–1704).
- Paredes, P., Gilad-Bachrach, R., Czerwinski, M., Roseway, A., Rowan, K., & Hernandez, J. (2014). PopTherapy: Coping with stress through pop-culture. In *Proceedings of the 8th international conference on pervasive computing technologies for healthcare* (pp. 109–117).
- Partala, T., & Surakka, V. (2004). The effects of affective interventions in human–computer interaction. *Interacting with computers*, *16*(2), 295–309.
- Patton, M. Q. (1990). *Qualitative evaluation and research methods*. SAGE Publications, inc.
- Perpetuini, D., Chiarelli, A. M., Cardone, D., Filippini, C., Rinella, S., Massimino, S., ... others (2021). Prediction of state anxiety by machine learning applied to photoplethysmography data. *PeerJ*, *9*, e10448.
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *12th international conference on evaluation and assessment in software engineering (EASE) 12* (pp. 1–10).
- Picard, R. W. (1997). *Affective computing*. MIT press.
- Picard, R. W., Vyzas, E., & Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE transactions on pattern analysis and machine intelligence*, *23*(10), 1175–1191.
- Pitkänen, H. (2012). *Voikko - free linguistic software and data for Finnish*. Retrieved 2019-09-24, from <https://voikko.puimula.org/>
- Plutchik, R. (1991). *The emotions*. University Press of America.
- Pons, G., & Masip, D. (2017). Supervised committee of convolutional neural networks in automated facial expression analysis. *IEEE Transactions on Affective Computing*, *9*(3), 343–350.
- Ramanujan, S., Scamell, R. W., & Shah, J. R. (2000). An experimental investigation of the impact of individual, program, and organizational characteristics on software maintenance effort. *Journal of Systems and Software*, *54*(2), 137–157.
- Revelle, W. (2011). An overview of the psych package. *Dep Psychol Northwest Univ*, *3*, 1–25.
- Romano, S., Caulo, M., Scanniello, G., Baldassarre, M. T., & Caivano, D. (2020). Sentiment polarity and bug introduction. In *International conference on product-focused software process improvement* (pp. 347–363).
- Rossiter, J. R. (2002). The C-OAR-SE procedure for scale development in marketing. *International journal of research in marketing*, *19*(4), 305–335.
- Rothstein, H. R., Sutton, A. J., & Borenstein, M. (2005). Publication bias in meta-analysis. *Publication bias in meta-analysis: Prevention, assessment and adjust-*

- ments, 1–7.
- Ruiz, M., Ramos, I., & Toro, M. (2001). A simplified model of software project dynamics. *Journal of Systems and Software*, 59(3), 299–309.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of personality and social psychology*, 39(6), 1161.
- Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological review*, 110(1), 145.
- Russell, J. A., & Barrett, L. F. (1999). Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. *Journal of personality and social psychology*, 76(5), 805.
- Salman, I., Rodriguez, P., Turhan, B., Tosun, A., & Gureller, A. (2020). What leads to a confirmatory or disconfirmatory behaviour of software testers? *IEEE Transactions on Software Engineering*.
- Salman, I., Turhan, B., & Vegas, S. (2019). A controlled experiment on time pressure and confirmation bias in functional software testing. *Empirical Software Engineering*, 24(4), 1727–1761.
- Schmidt, C. (2004). The analysis of semi-structured interviews. *A companion to qualitative research*, 253–258.
- Schreier, J. (2016, September). *The horrible world of video game crunch*. Retrieved 2021-10-26, from <https://kotaku.com/crunch-time-why-game-developers-work-such-insane-hours-1704744577> ([Online; posted 26-September-2016])
- Schulte, P., & Vainio, H. (2010). Well-being at work—overview and perspective. *Scandinavian journal of work, environment & health*, 422–429.
- Schutz, P. A., Pekrun, R., & Phye, G. D. (2007). *Emotion in education* (Vol. 10). Elsevier.
- Schwarz, N., & Clore, G. L. (1983). Mood, misattribution, and judgments of well-being: informative and directive functions of affective states. *Journal of personality and social psychology*, 45(3), 513.
- Scollon, C. N., Prieto, C.-K., & Diener, E. (2009). Experience sampling: promises and pitfalls, strength and weaknesses. In *Assessing well-being* (pp. 157–180). Springer.
- Shah, A. K., Mullainathan, S., & Shafir, E. (2012). Some consequences of having too little. *Science*, 338(6107), 682–685.
- Singh, P., & Suar, D. (2013). Health consequences and buffers of job burnout among indian software developers. *Psychological Studies*, 58(1), 20–32.
- Sommerville, I. (1996). Software process models. *ACM computing surveys (CSUR)*,

28(1), 269–271.

- Sonnentag, S., Brodbeck, F. C., Heinbokel, T., & Stolte, W. (1994). Stressor-burnout relationship in software development teams. *Journal of occupational and organizational psychology*, 67(4), 327–341.
- Spadini, D., Aniche, M., & Bacchelli, A. (2018). Pydriller: Python framework for mining software repositories. In *Proceedings of the 2018 26th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering* (pp. 908–911).
- Storbeck, J., & Clore, G. L. (2008). Affective arousal as information: How affective arousal influences judgments, learning, and memory. *Social and personality psychology compass*, 2(5), 1824–1843.
- Svenson, O. (1993). *Time pressure and stress in human judgment and decision making*. Springer Science & Business Media.
- Sykes, E. R. (2011). Interruptions in the workplace: A case study to reduce their effects. *International Journal of Information Management*, 31(4), 385–394.
- Tao, J., & Tan, T. (2005). Affective computing: A review. In *International conference on affective computing and intelligent interaction* (pp. 981–995).
- Thompson, B. (2004). *Exploratory and confirmatory factor analysis*. American Psychological Association.
- Tuomivaara, S., Lindholm, H., & Käsälä, M. (2017). Short-term physiological strain and recovery among employees working with Agile and Lean methods in software and embedded ICT systems. *International Journal of Human–Computer Interaction*, 33(11), 857–867.
- Van Berkel, N., Ferreira, D., & Kostakos, V. (2017). The experience sampling method on mobile devices. *ACM Computing Surveys (CSUR)*, 50(6), 1–40.
- Verduyn, P., Delaveau, P., Rotgé, J.-Y., Fossati, P., & Van Mechelen, I. (2015). Determinants of emotion duration and underlying psychological and neural mechanisms. *Emotion Review*, 7(4), 330–335.
- Wang, X., Xu, S., Peng, L., Wang, Z., Wang, C., Zhang, C., & Wang, X. (2012). Exploring scientists' working timetable: Do scientists often work overtime? *Journal of Informetrics*, 6(4), 655–660.
- Wanous, J. P., Reichers, A. E., & Hudy, M. J. (1997). Overall job satisfaction: how good are single-item measures? *Journal of applied Psychology*, 82(2), 247.
- West, S. G., & Hepworth, J. T. (1991). Statistical issues in the study of temporal data: Daily experiences. *Journal of personality*, 59(3), 609–662.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international*

- conference on evaluation and assessment in software engineering* (pp. 1–10).
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.
- Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. *Journal of Comparative Neurology*, *18*(5), 459–482.
- Zhang, J., Chen, Y., Gong, Q., Wang, X., Ding, A. Y., Xiao, Y., & Hui, P. (2020). Understanding the working time of developers in IT companies in China and the United States. *IEEE Software*, *38*(2), 96–106.

Original publications

- I Kuuttila, Miikka., Mäntylä, Mika, Claes, Maëlick and Elovainio, Marko (2017). Reviewing literature on time pressure in software engineering and related professions: computer assisted interdisciplinary literature review. *IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. pp.54-59. IEEE. DOI: 10.1109/SEmotion.2017.11
- II Mäntylä, Mika., Graziotin, Daniel and Kuuttila, Miikka.(2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Computer Science Review*, 27, pp. 16-32. DOI: 10.1016/j.cosrev.2017.10.002
- III Kuuttila, Miikka., Mäntylä, Mika., Farooq, Umar and Claes, Maëlick. (2020). Time Pressure in Software Engineering: A Systematic Review. *Information and Software Technology*, 121. Elsevier. DOI: 10.1016/j.infsof.2020.106257
- IV Kuuttila, Miikka., Mäntylä, Mika, Farooq, Umar and Claes, Maëlick. (2020). What Do We Know about Time Pressure in Software Development? *IEEE Software*. 38 (5). pp.32-38. IEEE. DOI: 10.1109/MS.2020.3020784
- V Claes, Maëlick., Mäntylä, Mika., Kuuttila, Miikka and Adams, Bram. (2018). Do programmers work at night or during the weekend?. In *Proceedings of the 40th International Conference on Software Engineering*, pp. 705-715. DOI: 10.1145/3180155.3180193
- VI Kuuttila, Miikka., Mäntylä, Mika. and Claes, Maëlick. (2020). Chat activity is a better predictor than chat sentiment on software developers productivity. *IEEE/ACM 5th International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. pp.39-43. DOI: 10.1145/3387940.3392224
- VII Kuuttila, Miikka., Mäntylä, Mika., Claes, Maëlick and Elovainio, Marko. (2018). Daily questionnaire to assess self-reported well-being during a software development project. *IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*. pp.39-43. DOI: 10.1145/3194932.3194942
- VIII Kuuttila, Miikka., Mäntylä, Mika, Claes, Maëlick., Elovainio, Marko and Adams, Bram. (2018). Using experience sampling to link software repositories with emotions and work well-being. *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. pp.29. ACM. DOI: 10.1145/3239235.3239245
- IX Kuuttila, Miikka., Mäntylä, Mika, Claes, Maëlick., Elovainio, Marko and Adams, Bram. (2021). Individual Differences Limit Predicting Well-being and Productivity Using Software Repositories: A Longitudinal Industrial Study. *Empirical Software Engineering*, 26(5), pp.1-30. Springer. DOI: 10.1007/s10664-021-09977-1

Reprinted with permission from The Institute of Electrical and Electronics Engineers, IEEE (I, IV, V), Association for Computing Machinery (VI, VII, VIII), and Elsevier (II, III). Paper IX is under Creative Commons 4.0 license.

Original publications are not included in the electronic version of the dissertation.

ACTA UNIVERSITATIS OULUENSIS
SERIES A SCIENTIAE RERUM NATURALIUM

749. Kainulainen, Tuomo (2020) Furfural-based 2,2?-bifurans : synthesis and applications in polymers
750. Varila, Toni (2020) New, biobased carbon foams
751. Tikka, Piiastiina (2020) Persuasive user experiences in behaviour change support systems : avoiding bottlenecks along the way to full potential of persuasive technology
752. Raulamo-Jurvanen, Päivi (2020) Evaluating and selecting software test automation tools : synthesizing empirical evidence from practitioners
753. Korhonen, Olli (2020) Service pathway personalization in digital health services
754. Moilanen, Antti (2021) Novel regulatory mechanisms and structural aspects of oxidative protein folding
755. Räikkönen, Jannikke (2021) Bone pathology in small isolated grey wolf (*Canis lupus*) populations
756. Heino, Matti (2021) Challenging DNA samples are valuable sources for genetic information of populations and individuals
757. Kujala, Valtteri (2021) Guidelines for building a combined e-commerce and ERP platform in micro-enterprises
758. Borshagovski, Anna-Maria (2021) Effects of social and visual environments on female sexual signaling
759. Vainionpää, Fanny (2021) Girls' choices of IT careers : a nexus analytic inquiry
760. Pyy, Johanna (2021) Forest management optimization according to nonlinear partial differential equation (PDE) and gradient based optimization algorithm
761. Mendes, Fabiana (2021) Insights from personality and decision-making in software engineering context
762. Karampela, Maria (2021) Recommendations to enable and sustain personal health data access and sharing : an empirical approach
763. Lämsä, Juho (2021) Behavioural mechanisms underlying food-deceptive pollination and neonicotinoid exposure of bumblebees
764. Mian, Salman Qayyum (2021) The social web as an ecosystem of networked improvement communities (NICS) : An interplay of user engagement, technology improvement, and the business opportunities as enablers

Book orders:
Virtual book store
<http://verkkokauppa.juvenesprint.fi>

S E R I E S E D I T O R S

A
SCIENTIAE RERUM NATURALIUM
University Lecturer Tuomo Glumoff

B
HUMANIORA
University Lecturer Santeri Palviainen

C
TECHNICA
Postdoctoral researcher Jani Peräntie

D
MEDICA
University Lecturer Anne Tuomisto

E
SCIENTIAE RERUM SOCIALIUM
University Lecturer Veli-Matti Ulvinen

E
SCRIPTA ACADEMICA
Planning Director Pertti Tikkanen

G
OECONOMICA
Professor Jari Juga

H
ARCHITECTONICA
Associate Professor (tenure) Anu Soikkeli

EDITOR IN CHIEF
University Lecturer Santeri Palviainen

PUBLICATIONS EDITOR
Publications Editor Kirsti Nurkkala



ISBN 978-952-62-3134-1 (Paperback)
ISBN 978-952-62-3135-8 (PDF)
ISSN 0355-3191 (Print)
ISSN 1796-220X (Online)