

<https://helda.helsinki.fi>

---

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

Kuutila, Miikka

2021-09

---

Kuutila , M , Mantyla , M , Claes , M , Elovainio , M & Adams , B 2021 , ' Individual differences limit predicting well-being and productivity using software repositories : a longitudinal industrial study ' , Empirical Software Engineering , vol. 26 , no. 5 , 88 . <https://doi.org/10.1007/s10664-021-09977-1>

---

<http://hdl.handle.net/10138/339685>

<https://doi.org/10.1007/s10664-021-09977-1>

---

cc\_by

publishedVersion

---

*Downloaded from Helda, University of Helsinki institutional repository.*

*This is an electronic reprint of the original article.*

*This reprint may differ from the original in pagination and typographic detail.*

*Please cite the original version.*



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

Miikka Kuuttila<sup>1</sup> · Mika Mäntylä<sup>1</sup> · Maëlick Claes<sup>1</sup> · Marko Elovainio<sup>2</sup> · Bram Adams<sup>3</sup>

Accepted: 12 May 2021 / Published online: 26 June 2021  
© The Author(s) 2021

## Abstract

Reports of poor work well-being and fluctuating productivity in software engineering have been reported in both academic and popular sources. Understanding and predicting these issues through repository analysis might help manage software developers' well-being. Our objective is to link data from software repositories, that is commit activity, communication, expressed sentiments, and job events, with measures of well-being obtained with a daily experience sampling questionnaire. To achieve our objective, we studied a single software project team for eight months in the software industry. Additionally, we performed semi-structured interviews to explain our results. The acquired quantitative data are analyzed with generalized linear mixed-effects models with autocorrelation structure. We find that individual variance accounts for most of the  $R^2$  values in models predicting developers' experienced well-being and productivity. In other words, using software repository variables to predict developers' well-being or productivity is challenging due to individual differences. Prediction models developed for each developer individually work better, with fixed effects  $R^2$  value of up to 0.24. The semi-structured interviews give insights into the well-being of software developers and the benefits of chat interaction. Our study suggests that individualized prediction models are needed for well-being and productivity prediction in software development.

**Keywords** Field study · Mining software repositories · Well-being · Experience sampling · Stress · Human factors · Negative result

## 1 Introduction

In recent years, the psychological well-being of software developers has drawn increased scientific interest from the fields of behavioral software engineering (Lenberg et al. 2015),

---

Communicated by: Daniel Méndez

✉ Miikka Kuuttila  
miikka.kuuttila@oulu.fi

Extended author information available on the last page of the article.

which borrows its name from the field of behavioral economics, and of “psychoempirical software engineering” (Graziotin et al. 2015). Software engineering researchers have established focused venues to study the affective states of software developers such as *The International Workshop on Emotion Awareness in Software Engineering*.<sup>1</sup>

Subjective well-being has been described as a broad range of phenomena, including people’s emotional responses, domain satisfactions, and global judgments of satisfaction (Diener et al. 1999). People have been shown to use momentary affective states as information in judging their well-being (Schwarz and Clore 1983). Core affect has been defined as “A neurophysiological state that is consciously accessible as a simple, nonreflective feeling that is an integral blend of hedonic (pleasure–displeasure) and arousal (sleepy–activated) values”, and “the simplest raw (nonreflective) feelings evident in moods and emotions” (Russell 2003). Studies on mining software repositories have made several recent attempts to build tools and to reason about the affective states of software developers by utilizing sentiment analysis (e.g., Mäntylä et al. 2017 and Novielli et al. 2018). However, to the best of our knowledge, no prior studies have attempted to link daily experience sampling of affective states with measures from software repositories in a longitudinal industrial setting. According to Scollon et al. (2009), strong points in experience sampling are its ability to document real-life experiences that improve ecological validity, to reduce of memory bias, and to augment of other research methods. We give more details of experience sampling in Section 2.3.

This paper investigates whether different software development actions are associated with different affective states and self-reported well-being. To achieve our goal, we used experience sampling methodology and created a questionnaire to be completed daily in an industrial software project setting. The questionnaire is based on psychosocial theories of work (Karasek 1990). The questionnaire assesses hurry, stress, sleeping problems, interruptions, ineffective software development (defined as poorly working tools, processes or communication), and job control (independence). Metrics obtained with the questionnaire were then linked to measures obtained from software repositories related to code commit activity, amount of social interaction in an instant messaging application, the sentiment expressed through words, emoticons and emojis, and job events. We built generalized linear mixed effects models to understand the relations between software repository variables, which reflect software development actions, and the answers to the questionnaire. Additionally, we conducted semi-structured interviews to better understand the project context and reasons for discovering different relationships in the models.

Hence, our research questions were formulated as:

- RQ1 Does everyone in the development team share the same level of well-being?
- RQ2 Can software developers’ actions predict well-being?\*
- RQ3 Can software developers’ well-being and actions predict software developers’ productivity?\*
- RQ4 Can interviews give further information about experienced well-being of software developers?+

In our research questions, experienced well-being refers to our questionnaire. Our questionnaire asks developers for stress, hurry, sleeping problems, interruptions, independence, and ineffective software development. Similarly, in our research questions, software developers’ actions refer to the multitude of variables mined from software repositories, e.g.,

---

<sup>1</sup><https://semotion.github.io/2021/>

commit related activity, amount of communication, expressed sentiment in communication, and job events.

This paper is an extension of our prior conference paper (Kuutilla et al. 2018b) that analyzed RQ2 and workshop paper that investigated RQ3 (Kuutilla et al. 2020b) with different methodologies. Kuutilla et al. (2018b) looked at a limited set of count variables related to productivity and chat messages and their relationship to the questionnaire variables with logistic regression, where we also used binning for these count variables. Here we extend with more variables like sentiment analysis, customer meeting, and build failure information. Additionally, we changed the statistical analysis to a generalized linear mixed-effects model with an autocorrelation structure. This allows us to control the effect at the individual level while also accounting for the autocorrelation. Kuutilla et al. (2020b) examines the sentiment analysis related variables in relation to lines of code and commits produced by the developers. Here we add the questionnaire responses, customer meetings and build failure information. Additionally, we use generalized linear mixed-effects with autocorrelation structure to control the effect of the individual.

For clarity, we have marked the research questions with added variables and new statistical analysis using “\*” in the above description. Semi-structured interviews were completely new for this extension and marked with a “+” for this reason.

The rest of the paper is structured as follows. The relevant background from psychology and software engineering is introduced in Section 2. The methodology for creating the daily questionnaire and executing this study is explained in Section 3. In Section 4 we present the results to our research questions and discuss them in Section 5. We discuss internal and external threats in Section 6. Lastly, conclusions are provided in Section 7.

## 2 Background

### 2.1 Work Well-Being in Psychology

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 et al. 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 the societal level.

Very broadly, stress can be defined as a state of real or perceived disharmony that threatens homeostasis, i.e., a state in which equilibrium and optimal functioning, including body temperature of an organism, are threatened, by either intrinsic or extrinsic forces, i.e., stressors (Chrousos and Gold 1992). Various physiological correlates to stress include blood pressure, heart rate, and galvanic skin response (Vrijkotte et al. 2000; Schuler 1980). Prolonged stress can lead to cognitive impairments (McEwen and Sapolsky 1995), and neuronal disturbances resembling changes that are observed in the brain during depression (De Kloet et al. 2005).

A multitude of definitions for stress in organizational settings are collected and discussed by Schuler (1980). The author concludes that these definitions “suggest that individuals are ‘under stress’ particularly when the demands of the environment exceed (or threaten to exceed) a person’s capabilities and resources to meet them or the needs of the person are not being supplied by the job environment.”

In more recent times, the job demands-resources model (Karasek 1990; Bakker and Demerouti 2007) is commonly used to explain employee well-being. The model generally divides job-related factors into two categories: demands and resources. Well-being is the outcome of the balance between these two categories, while job strain is produced by an imbalance between job resources and demands. Resources can be divided into personal and job resources. Personal resources are positive self-evaluations linked to resiliency and a sense of ability to control and impact upon the environment. On the other hand, job resources are physical, social, psychological and/or organizational aspects that are functional in achieving work goals, reducing demands, and stimulating personal growth (Xanthopoulou et al. 2009). Evidence suggesting job resources, personal resources, and work engagement are reciprocal over time, and support employee well-being exists (Xanthopoulou et al. 2009). Similarly, evidence of worker autonomy and social support increasing work engagement exists (Taipale et al. 2011). Work demands and continued job strain are connected to exhaustion and burnout (Demerouti et al. 2001; Xanthopoulou et al. 2007). Related to software development, the usage of information and communication technology is seen to be one possible source of stressors (Tarafdar et al. 2007).

## 2.2 Work Well-Being and Emotions in Software Engineering

Sonnentag et al. (1994) surveyed 180 software developers to identify factors related to burnouts, they discovered 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.

Kuutila et al. (2020a) reviewed the effects of time pressure on software productivity and quality. The evidence shows lessened quality due to time pressure, while the evidence on productivity is two-fold: most cost and scheduling models assume increased total effort with compressed schedules, but empirical studies and experiments report increased efficiency under time pressure.

Fucci et al. (2018) investigated the effect of sleep deprivation on software developers and found that even a single night of sleep deprivation had a negative effect on software development quality. However, in a different study, it has been noted that two-thirds of developers work during normal working hours, while large differences between projects exist (Claes et al. 2018b).

Interruptions and their effects on software development work have been investigated. Tregubov et al. (2017) showed that developers working in multiple projects use a significant amount of their working time on context switching. Sykes (2011) discovered that senior developers and technical leads were experiencing more interruptions in their work in comparison to the regular staff at a software development company, guidelines on avoiding interruptions for software developers are also given in the work. Brumby et al. (2019) has synthesized studies on interruptions' effects on productivity in software engineering to insights, some of which concern the types of interruptions. The most relevant insights to our study include "Shorter interruptions are less disruptive than longer interruptions" and "Interruptions can cause stress, particularly e-mail interruptions."

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). From software engineering context, Jongeling et al. (2015) compared and evaluated general sentiment analysis tools and their performance 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.

There are a limited number of studies on the usage of emoticons by software developers, but Claes et al. (2018a) have studied the use of emoticons by developers in two issue trackers. They found out project-level differences between Apache and Mozilla projects. Moreover, there were also differences between geographical locations, with developers from Europe and northern America using more emoticons.

With consideration of the pertinent literature, the novelty of our work lies in combining multiple data sources (experience sampling and repository mining), and examining the links between these data sources using multivariate models.

## 2.3 Experience Sampling Method (ESM)

### 2.3.1 Overview from Psychology

The experience sampling method (ESM), also known as the daily diary method, studies everyday experiences and behavior in a natural environment, with data gathered both from both psychological and physiological sources (Alliger and Williams 1993). The strengths of ESM lie in its empirical nature in which documentation of real-life experiences increase its ecological validity, its allowance of investigating within-person processes, its reduction of memory bias compared with other methods using self-reports, its allowance of investigating contingent behavior, and its ability to augment other research methods. Among possible weaknesses related to experience sampling are the self-selection bias, motivation issues in the acquired sample, the limited number of questions in data gathering, and the possible reactivity to the research setting (Scollon et al. 2009).

Experience sampling methods have been divided into three categories (Scollon et al. 2009) based on the time when the experiences are gathered: interval-contingent sampling, event-contingent sampling and signal-contingent sampling. Interval-contingent sampling refers to collecting data after a given time interval (e.g., hourly, daily, or weekly). In event-contingent sampling, data are gathered after specific events (e.g., after every meeting or social interaction). Lastly, signal-contingent sampling refers to a situation where participants in the study are prompted to answer at a randomly timed signal. A variety of devices can be used to remind subjects to respond to surveys and questionnaires, such as personal digital assistants, booklets, beepers, or wristwatches (Kimhy et al. 2006). However, reminders via email or SMS are also commonly applied.

In previous studies on work well-being outside of software engineering, experience sampling methods and daily questionnaires have been used to study events, moods and behavior in a work setting. Some examples of the findings are that negative job events are five times more likely to be related to a negative mood than positive job events are to a positive mood (Miner et al. 2005). Additionally, job satisfaction has been measured with experience sampling methodology and evidence has been found that affect and cognition are antecedents to job satisfaction (Ilies and Judge 2004). Continued cognitive engagement, more positive affect during work activity than during leisure activity and preference for work activities over leisure activities have been linked to workaholism in an ESM study (Snir and Zohar 2008). Outside the work context, experience sampling has also been used to study interaction with information systems. For example, it has been found that an increase in the usage of Facebook predicted a lower life satisfaction level (Kross et al. 2013). The novelty of our work is to combine ESM data with data acquired from software repositories.

### 2.3.2 Challenges in Statistical Analysis

Experience sampling methods produce time-series data that should be considered during analysis. As some statistical tests assume the independence of observations, non-independence in the time series data gathered with experience sampling is a problem needing action. West and Hepworth (1991) identify three main sources of non-independence that can occur in the data: auto-correlation, trend, and seasonality, all of which should be accounted for in an analysis.

Repeated measures over time can create auto-correlation, i.e., time-dependent data in violation of the assumption of independence. For example, the level of stress felt today is not completely independent on the level of stress felt yesterday. Controlling for the trend is important when cross-correlating time series, as underlying trends, create spurious correlations between the time series. For example, an increasing trend in the number of software engineers over time would create spurious correlations with many software engineering output measures such as commits and defect reports. The seasonality components usually refer to daily, weekly, monthly or yearly cycles, for example, stress levels could be perceived as stronger on Mondays.

### 2.4 Negative Results

Publication bias “is the tendency on the parts of investigators, reviewers, and editors to submit or accept manuscripts for publication based on the direction or strength of the study findings” (Dickersin 1990). Publishing negative results have been seen to fight publication bias (Dirnagl and Lauritzen 2010). Still, evidence points toward decreased publishing of negative results in modern times (Fanelli 2012).

In software engineering there is also increased interest in allowing negative results to break the publication bias barrier. Related to our work, a couple of negative results have been published. Both 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 (Jongeling et al. 2017; Lin et al. 2018).

## 3 Methodology

An experience sampling study was conducted in a medium-sized software company in Finland. During our study, it employed four to five hundred people across its projects.

We developed a questionnaire that was sent to one of their teams developing a service with Agile methods and continuous delivery. Some elements from Scrum were present, the development process had iterations, after which results were presented, and future directions were planned in a retrospective. Tasks were tracked in a “kanban” board style with tickets: from there, they were completed as ready when they were deployed and tested to the staging environment. The project has a single customer and meetings with the customer were held almost weekly. The software is used in the daily operations of the customer, but it is not safety-critical software.

### 3.1 Daily Questionnaire

We constructed a short questionnaire, which was answered daily by the project team from the software company. The goal of this questionnaire was to produce data related to the

work experience of the software project personnel, specifically developers. We piloted the questionnaire with the authors of this article. The aim was to produce a questionnaire that can be taken quickly, to achieve high response rates across a prolonged period. Therefore, we used single-item measurements, which have been shown in general to produce valid data in prior studies (Wanous et al. 1997; Nagy 2002; Elo et al. 2003).

The questionnaire was constructed by picking relevant items from the survey done by Elovainio et al. (2015), that studied work well-being of physicians and was published in the *Journal of Occupational Health Psychology*. The questionnaire includes six single items that measure variables related to job well-being on an ordinal five-point scale. Thus, our questions represent theoretical concepts related to work health and well-being, as explained in Section 2. As the past survey was not done in the software engineering domain, we added one software engineering specific item to the questionnaire. Only one software-specific question was added to the questionnaire in order not to overload the respondents. This resulted in the following statements (without the emphasis) included in the questionnaire:

- *I can make independent decisions in my work.* Individuals' **independence** and autonomy have been under study as a mediating factor between job demands and resources (Bakker et al. 2005; Xanthopoulou et al. 2007), i.e., there is evidence that increased autonomy in work tasks lessens the effects of job demands such as time pressure.
- *I am in a hurry and have too little time to finish the task properly.* **Hurrying** to complete work, also known as time pressure, is a job demand, and has a complex relationship with performance (Bakker and Demerouti 2007; Kuuttila et al. 2020a). It has been shown to be associated with increased performance in the short term (Nan and Harter 2009; Mäntylä et al. 2014), but also higher stress (Svenson 1993) and even burnout (Donald et al. 2005; Bakker et al. 2005; Sonnentag et al. 1994).
- *I feel interrupted while working.* **Interruptions** to work increase the effort needed for task completion and have also been shown to increase time pressure and stress in the software development context (Mark et al. 2008). The types of interruptions also play a role, with longer interruptions being worse for performance (Brumby et al. 2019).
- *I experience ineffective software development (poor processes, poorly performing tools or poor communication with the development team).* This question includes common topics related to productivity in software processes (Diaz and Sligo 1997), tools (Bruckhaus et al. 1996), and communication (Wagner S and Ruhe 2018).
- *I feel stressed (refers to a situation in which the respondent feels tense, restless, nervous, or anxious).* In our case this refers to distress. **Stress** is modeled to be the result of an imbalance of demands and resources (Bakker and Demerouti 2007), it has been linked to cognitive impairments (McEwen and Sapolsky 1995), and affective states related to depression (De Kloet et al. 2005).
- *I experience sleeping problems (difficulty in falling asleep or waking up several times during the night).* **Problems sleeping** have been strongly linked to stress and increased job demands (Åkerstedt et al. 2002; Linton 2004).

As previously stated, the questionnaire was constructed by picking relevant items from the survey done by Elovainio et al. (2015). We did not opt for multiple items, that is multiple questions measuring the same variable. This is because the developers answering several dozens of questions daily would not have been practical nor possible. Our single 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 regarding stress is originally from the general health questionnaire "GHQ-12"



(Goldberg and Blackwell 1970) and refers to distress; and lastly the question concerning sleeping problems is from the Jenkins scale (Jenkins et al. 1988). These questions were slightly modified to fit our five-point scale: the respondents were asked to rate these six items with the question: “How frequently has the following condition occurred since the last time you answered this survey?”. These items were then ranked on a five-point scale. 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, we met with the project personnel to explain the purpose of the study, and to clarify why daily answers were needed for the questionnaire.

The developed questionnaire was sent to the developers of the project over a period of 8 months (from April 10th, 2017 to January 12th, 2018). We used Webropol<sup>2</sup> to send the questionnaire every working day by email at 8 a.m. and to collect the responses. Developers who moved from or to another project, or started working in multiple projects at the same time, stopped answering the questionnaire. Developers with less than ten responses were discarded from the data analysis.

For data analysis, a total of 526 responses were received from eight respondents. All responses included answers to all questions. None of the answers were preset, i.e. there was no pre-checked default answer. Developers could also simply not answer the questionnaire sent to them during some of the days. Multiple answers received during the same day by one individual were replaced with the mean of those answers, reducing the number of analyzable answers to 502. We also received another five answers during a weekend, and we removed these answers from analysis, further reducing the answers to 497. Considering the summer holidays, the total response rate is 37,5% (526 / 1404) for eligible respondents. Looking at response times during the day before aggregating multiple answers, around 68,5% were given between 7:00–10:00 a.m., and around 95% during normal sliding working hours of 7:00–16:00. Two answers were given before 7:00 a.m., and a total of 19 after 5:00 p.m. The response rate was the highest during the first three months of the study (58% of the total responses), decreasing steadily afterward with the last three full months having 23% of total responses.

## 3.2 Mining Software Repositories

In Table 1, we provide the name and a short description of all the variables acquired from the software repositories. In the following subsections, we explain why and how these variables were acquired.

### 3.2.1 Version Control System

We used Perceval (Dueñas et al. 2018) to extract the list of commits from the Git repository used by the project team. For each day of the period during which the developers answered the questionnaire, we computed for each respondent the number of commits made (*ncommits*) and the number of lines of code modified (*nloc*). While software development contains tasks not captured by these metrics, the number of commits and lines of code have been widely used as proxy measures for productivity in software engineering (Mockus et al. 2002; Boehm and et al. 1981). Recent work has noted lines of code having the highest correlation with self-evaluated productivity (Murphy-Hill et al. 2019).

---

<sup>2</sup><http://w3.webropol.com/start/>

**Table 1** Overview of software repository variables. All variables per day

Variable	Description	Dimension
v1 commits	#commits each developer made	Code Activity
v2 loc	# of code lines changed by the developer	Code Activity
v3 filelogsum	#files changed by the developer in all commits during the day	Code Activity
v4 nchat	#chat messages sent	Amount of Communication
v5 emoticon	% of messages sent by the developer with an emoticon or an emoji	Expressed sentiment
v6 joypr	% of messages sent by the developer with joy emoticon or emoji	Expressed sentiment
v7 sadconfusionsurprisepr	% of messages sent by the developer with an emoticon or an emoji expressing sadness, confusion or surprise	Expressed sentiment
v8 negative valence	% of messages sent by the developer with words with negative valence scores	Expressed sentiment
v9 positive valence	% of messages sent by the developer with words with positive valence scores	Expressed sentiment
v10 minimum valence	The minimum score of the valence word for that day	Expressed sentiment
v11 maximum valence	The maximum score of the valence word for that day	Expressed sentiment
v12 low arousal	% of messages sent by the developer with words with low arousal scores	Expressed sentiment
v13 high arousal	% of messages sent by the developer with words with high arousal scores	Expressed sentiment
v14 minimum arousal	The minimum score of the arousal word for that day	Expressed sentiment
v15 maximum arousal	The maximum score of the arousal word for that day	Expressed sentiment
v16 meeting	binary variable of whether there was a meeting with the customer during the day	Job event
v17 failure	#times production tests failed during the day	Job event

Entropy has been used to quantify the complexity of code changes in previous literature (Hassan 2009). However, we decided to use the number of files changed by the developer each day, without considering the size of the project itself. This is because the number of developers grew during the project, some of whom did not answer the questionnaire. Result is the variable *filelogsum*, which describes the number of times files were changed by the developer during the day, transformed to the base-10 logarithmic scale, as a result of the skewed nature of the distribution.

### 3.2.2 Mining Chat Messages

Additionally, the company also provided us with a JSON dump of the chat room used by the developers. The specific tool used for communication changed during our study from

Hipchat<sup>3</sup> to Slack.<sup>4</sup> From this chat archive, we computed the daily number of chat messages (*nchat*) for each respondent.

We also translated lexicons used in the software engineering context for measuring arousal (Mäntylä et al. 2017) and valence to Finnish to do rudimentary sentiment analysis on the chat logs. Chat logs were lemmatized using the open-source software Voikko (Pitkänen 2012), and then scored on valence and arousal using the translated lexicons. The arousal or valence scores in the lexicons range from 1 to 9, and thus are centered around 5. Hence low valence and arousal are shown in scores under 5, and high valence and arousal in scores over 5. We use this information in the variables *negative valence*, *positive valence*, *low arousal* and *high arousal*. Hence, the variable *negative valence* contains the percentage of messages containing at least one word with a valence score below 5 and the variable *positive valence* denoted as the percentage of messages containing at least one word with a valence score above 5. The same method was applied to for variables *low arousal* and *high arousal*. Similarly, we also calculated the maximum and minimum arousal and valence scores for each day for each developer, and these are found in the variables *minimum valence*, *maximum valence*, *minimum arousal*, and *maximum arousal*.

We also extracted emoticons and emojis that were used in the chat messages. Emoticons are textual representations of human emotion using only keyboard characters such as letters, numbers, or punctuation marks. Emojis refer to "picture characters" or pictographs (Miller et al. 2016). Similar to some of the authors' previous work (Claes et al. 2018a), we manually classified the emoticons to the basic emotions of Plutchik's wheel of emotions (Plutchik 1991): joy, sadness, surprise, confusion, and anger. The used list of emoticons and emoji, and their associated emotions, is available online.<sup>5</sup> The first and third authors classified the emoticons and achieved a 79.5% agreement with a Cohen's kappa of 0.7, after which we went through the cases where we disagreed. With these emoticons, we calculated the percentage of messages containing *emoticons* and emojis, the percentage of messages containing emoticons and emoji *related to joy*, and the percentage of messages containing emoticons and emojis *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*. For conciseness in the results section of this manuscript, emoticons refer to both emoticons and emojis.

### 3.2.3 Factor Analysis and Measurement Model

We used factor analysis to study the structure of the underlying variables in our data set (Thompson 2004). We explored the data sources from Table 1 with the *fa.parallel* function<sup>6</sup> for the optimum number of factor, then we used the *fa* function<sup>7</sup> to find the minimum residual (minres) solution using 100 iterations. The resulting factors are in the left side of Fig. 1.

For these factors we computed the goodness of fit statistics, which shows a very good fit (Table 2). For choosing the goodness of fit statistics, we followed the figure given by Sun (2005) and give sample based goodness of fit indices Tucker Lewis index (Tucker and

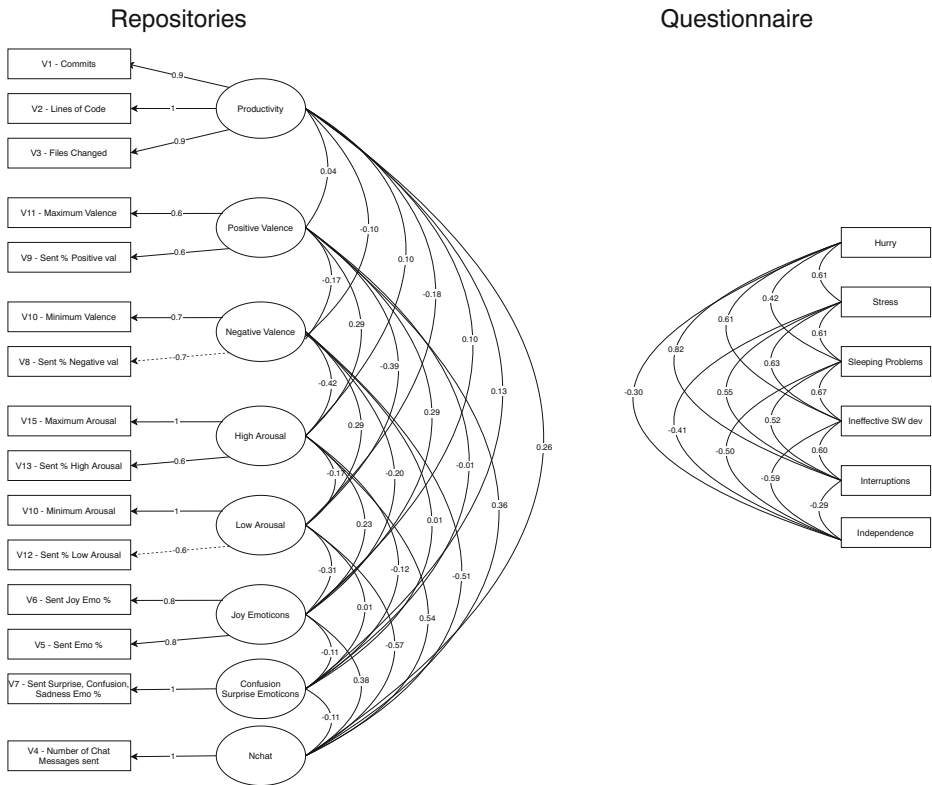
<sup>3</sup><http://www.businessinsider.com/atlassian-launches-hipchat-successor-stride-2017-9>

<sup>4</sup><https://slack.com/>

<sup>5</sup><https://github.com/M3SOulu/autotime-esm>

<sup>6</sup><https://www.rdocumentation.org/packages/psych/versions/2.0.8/topics/fa.parallel>

<sup>7</sup><https://www.rdocumentation.org/packages/psych/versions/2.0.8/topics/fa>



**Fig. 1** The measurement model resulting from factor analysis, showing variable weights to factors and correlations between latent variables used in our study

Lewis 1973) and the root mean square residual (RMSR). The TLI or Non-Normed Fit Index is a fit measure comparing the fit in relation to the null model (Marsh et al. 1996). RMSR is a descriptive fit defined as “is defined as the square root of the mean of the squared fitted residuals” (Schermelleh-Engel et al. 2003). Measures of TLI and RMSR indicate a very good fit. While unusual, the TLI can have values greater than one, see discussions by Anderson and Gerbing (1984) and by Muthén and Muthén (2017).

Our measurement model (Fig. 1) shows the relationships between latent variables and their indicators (Bollen 2001). On the left factors created by exploratory factor analysis are shown, on the right correlations between variables acquired with the questionnaire are

**Table 2** Goodness of fit statistics for factors discovered with exploratory factor analysis

Factor	TLI	RMSR
Productivity	0.99	0.01
Positive Valence	1.029	0.01
Negative Valence	1.01	0
High Arousal	1.01	0
Low Arousal	1.01	0
Joyemo	1.01	0

shown. The oval shapes under “Repositories” denote the factors from Table 2. The rectangles under “Repositories” show the variables from Table 1. Lines between variables and factors show weights, with dotted lines signaling negative weights. On the

### 3.2.4 Generalized Linear Mixed Effect Models

We used generalized linear mixed effects models as they can be used to study both fixed and random effects. We used the package `nlme` (Pinheiro et al. 2017) to construct the models because it can easily accommodate auto-correlation structures. The variables specified in our measurement model were evaluated as fixed effects. For random effects, we used a unique respondent identifier, variable specifying the day of the week (“weekday”), and a time variable designating the day during the study (i.e. the first day of the study as 1, the second as 2, and so on). We used the function `r.squaredGLMM` from the package `MuMIn` (Barton 2009) to calculate both the marginal and the conditional  $R^2$  values. Marginal  $R^2$  values represent the variance explained by the fixed effects, while the conditional  $R^2$  values are interpreted as a variance explained by the entire model, including both fixed and random effects. Calculating marginal and conditional  $R^2$  values for mixed effects models is based on the work of Nakagawa and Schielzeth (2013). When constructing models for individuals, we took the four respondents with the highest number of answers to the questionnaire.

### 3.2.5 Seasonality and Auto-correlation

We studied the trends and seasonality in our collected data with the R function `decompose`<sup>8</sup> and found weekly seasonality for all the software repository variables. The weekly seasonality of the chat messages is the highest. The average number of chat messages sent on Mondays were 30.7, 25.8 on Tuesdays, 30.8 on Wednesdays, 41.6 on Thursdays, and 45.7 on Fridays. By comparison, the seasonality of commits per day is weaker. The average commits on Mondays was 9.7, 7.8 on Tuesdays, 7.8 on Wednesdays, 11.2 on Thursdays, and 8.5 on Fridays. To account for time-series data and control for weekly seasonality in the data, we added a weekday variable as a random effect to the models.

We also investigated the auto-correlations of the data with the `acf` function<sup>9</sup> of the forecast R package (Hyndman et al. 2007), and found strong auto-correlations for all the questionnaire variables. As a consequence, we added these variables as random effects to our generalized linear mixed effects of models. In practice, this means we used the `corARMA` function<sup>10</sup> with a 10 day moving average structure when making general models, and a 5 day moving averages when creating models for individuals. We observed no meaningful differences between results in our models based on the used auto-correlation structure whether it was a one day average or different moving averages, but we had troubles with model convergence depending on the used auto-correlation structure. Convergence problems are related to the complexity of random effects, and are further discussed in Section 6.

## 3.3 Semi-structured Interviews

Semi-structured interviews have for long been advocated for and used to collect qualitative data about phenomena in software engineering (Hove and Anda 2005). Semi-structured

<sup>8</sup><https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/decompose>

<sup>9</sup><https://www.rdocumentation.org/packages/forecast/versions/8.3/topics/Acf>

<sup>10</sup><https://www.rdocumentation.org/packages/nlme/versions/3.1-137/topics/corARMA>

**Table 3** Interviewee characteristics

ID Role	#Years in SW industry	Degree
PM/Project Manager	13 years	Masters
Developer 1	7 years	Masters
Developer 2	4 years	Masters

interviews have been recommended as a supplement to surveys and questionnaires, when important questions remain after collecting quantitative data (Adams 2015). Hence, after publishing our previous work (Kuuttila et al. 2018b), from which this paper is based, we conducted semi-structured interviews to answer “why questions” about the quantitative data we had gathered. The interview questions were thus designed to better explain our prior results, and the translations of these questions are available at Github.<sup>11</sup>

For conducting and designing the interviews, we followed the guidelines given by Adams (2015). Drafting interview questions and making the interview guide was done collectively by the authors, aiming for open-ended questions for which we could ask follow-up questions and query clarifications. Since visualizing timelines and results have been advocated for project-level retrospectives Bjarnason et al. (2014), in our case, we decided to help remembrance and recollection by visualizing the answers to the questionnaire by sending graphs of individual-level responses to the interviewees before the interview. The first author interviewed the project manager and two developers for this study. One interview was done in person and two over a video call. The interviews totaling almost three hours were recorded with written permission from the interviewees and transcribed verbatim after the interview process.

We present some background information on the interviewees in Table 3. The project manager was not part of the data collection of the quantitative research questions, but we believed project managers’ views on the project were valuable.

The interview started with questions about the questionnaire procedure itself, and about the individual level graphs we mentioned previously. The main aim of this was to help the respondent recall the answering period, but also to get any recommendations to make the procedure easier and to increase the response percentage. After this, we asked a question related to the results of the previous conference paper Kuuttila et al. (2018b), and the kind of explanations the respondents might have. Next, we asked questions related to the new variables we were bringing to the analysis: expressed sentiment, emoticon usage, and job events. Related to job events, we asked for any recollections of hurry and stress periods during the questionnaire.

We followed the analytical strategy of Schmidt (2004) for analyzing the transcripts of the semi-structured interviews. As advocated by Schmidt (2004), the analysis process comprised repeated and intensive reading of the transcriptions and the development of a coding scheme from analytical categories. In the end, our very simple scheme contained three different codes: (1) facilitating activities helping well-being; (2) barriers to well-being and (3), explanations of our results. The fourth step of quantifying the material involved mainly finding codes that were uniform and coherent across the interviewees. These have been mentioned in the results in the form of whether the respondent agreed or disagreed on topics, and emphasizing themes that were uniform across the three interviews. Not mentioning a particular topic was not interpreted as a disagreement.

<sup>11</sup><https://github.com/M3SOulu/autotime-esm>

**Table 4** Inter-coder agreement of the respondents

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

## 4 Results

### 4.1 RQ1—Does Everyone in the Development Team Share the Same Level of Well-Being?

Our main motivation for this research question was to understand how well-being was felt in the software project under study. In particular, if several individuals on the development team reported similar well-being and affective states simultaneously during the development project, it could be that external demands such as deadlines could affect the whole development team at the same time. For example, there is evidence that part of work-related stress is shared within organizations (Semmer et al. 1996). Additionally, related work on time pressure has called for organizational-level studies (Silla and Gamero 2014).

The values produced by Krippendorff's alpha (Krippendorff 2011) are between 1 (perfect agreement), 0 (statistically unrelated), and -1 (perfect disagreement). To interpret the values, Krippendorff proposed thresholds (Krippendorff 1980), where a value of 0.2 is considered poor, and values greater than 0.7 are good. We observe poor disagreement between respondents for all questionnaire variables. Table 4 shows values from -0.214 for Sleeping Problems up to -0.099 for Hurry. These negative values could imply two things: either the respondents feel each affective state individually rather than on a group level, or they use different calibrations of the scales. That is, some individual respondents consider a value of 2 for Hurry normal while others consider it to be exceptional.

### 4.2 RQ2—Can Software Developers' Actions Predict Well-Being?

As developers' responses to the questionnaire differed at the same time points, we analyzed them in relation to several factors derived from software repositories with a one-day time lag while taking the individual into account. For analysis, we chose generalized linear mixed models, and we use all the predictors due to the exploratory nature of our study. In Tables 5, 6, 7, 8, 9, 10 and 11 we investigate the relationship from software repositories to our questionnaire responses with a one-day lag, i.e. can the previous days repository metrics predict current days questionnaire responses. As the questionnaire was sent each morning and most of the responses were also given in the morning (see the end of Section 3.1), using previous days' repository data seemed most reasonable to us.

In generalized linear mixed effect models (Section 3.2.4), the marginal  $R^2$  value represents the variance explained by the fixed effects, while the conditional  $R^2$  value is interpreted as a variance explained by the entire model, including both fixed and random effects. We also provide a null model conditional  $R^2$  value. They refer to models where only random variables are used to explain the predicted variable. Random variables in our case

**Table 5** 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**

Predicted	Hurry	Stress	Sleep	Inter- ructions	Ineffe ctive	Indepe ndence
prod	0.07 ( <b>0.02</b> )	0.19 (< <b>0.001</b> )	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 (< <b>0.001</b> )
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 ( <b>0.01</b> )	0.09 ( <b>0.03</b> )	-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 ( <b>0.05</b> )	0.44 (0.25)	-0.20 (0.61)	0.28 (0.38)	0.31 (0.18)	0.35 ( <b>0.03</b> )
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 ( <b>0.001</b> )	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	0.74	0.39	0.44	0.67	0.80	0.66
C. $R^2$						

were the respondent ID, and the auto-correlation variables weekday and date as a number from 1 to 240.

Table 5 shows the models predicting questionnaire answers with the previous working days repository variables. In other words, do the actions of the previous working day predict well-being and the answer on the questionnaire to be completed the following morning. As we can see in Table 5 the conditional  $R^2$  value is considerably higher than the marginal  $R^2$  value in every model, meaning random effects explain overwhelmingly more variance in the predicted variable than fixed effects. When looking inside random effects, the respondent ID variable is mostly behind the dominant conditional  $R^2$  value. This means that the individual in question has the highest effect on the prediction of the questionnaire variable.

The highest effect of fixed effects can be found for prediction stress, with a marginal  $R^2$  value of 0.02. For the fixed effects in the other models predicting questionnaire variables, a marginal  $R^2$  value of 0.01 or less is found by the models.

Although the marginal  $R^2$  value is small, we go through the statistically significant regression coefficients as they may spark future works. When looking at the predictors in fixed effects, the highest coefficient is in productivity when predicting stress ( $p < 0.001$ ). Productivity was also a significant predictor of a hurry. In other words, the developer’s previous day’s higher productivity was associated with experiencing hurry and stress the next day. We also found that expressing positive valence (pval) was associated with increased independence the next day but so was using sad or confused emoticons (scsemo). Therefore, it may be that independence is increased in both expressing positive and negative emotions. Expressing elevated arousal (har) was associated with developers reporting more



**Table 6** 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**

Variable:	DevA	DevB	DevC	DevD
prod	0.11 (0.07)	-0.53 (< <b>0.001</b> )	0.08 (0.47)	
nchat	-0.02 (0.80)	0.15 (0.23)	-0.09 (0.51)	
pval	-0.06 (0.56)	-0.51 (< <b>0.001</b> )	-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 ( <b>0.03</b> )	
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	

interruptions and ineffective software development. Finally, we find that meetings reduced the feeling of hurry the next day.

Because the effect of the respondent was very high when predicting questionnaire outcomes, as seen in Table 5, we constructed models with the data from four the individuals with the highest number of responses to the questionnaire. Tables 6–11 show models predicting questionnaire answers with a single individual's data. Similar to Table 5, the questionnaire answers were predicted using the previous day's repository variables. Empty columns in the tables mean that the model did not converge. We also gave a null model conditional  $R^2$  value, the amount of variance predicted solely by random effects, which in

**Table 7** Generalized linear mixed models predicting stress for the next day with today's repository variables. A p-value of 0.05 or less is denoted in **bold**

Variable:	DevA	DevB	DevC	DevD
prod	0.38 ( <b>0.01</b> )		0.15 (0.23)	
nchat	-0.25 (0.16)		-0.39 ( <b>0.02</b> )	
pval	-0.13 (0.62)		0.17 (0.49)	
nval	-0.19 (0.28)		0.19 (0.32)	
har	0.24 (0.34)		-0.30 (0.19)	
lar	-0.18 (0.31)		0.08 (0.64)	
joyemo	0.02 (0.75)		-0.03 (0.81)	
scsemo	0.97 (0.30)		-0.14 (0.91)	
meeting	-0.03 (0.89)		-0.13 (0.53)	
failure event	0.25 (0.10)		0.09 (0.55)	
Marginal $R^2$	0.10		0.09	
Conditional $R^2$	0.10		0.09	
Null Model C. $R^2$	<0.01		0	

**Table 8** 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**

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	<b>0.54 (0.02)</b>	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 ( <b>0.04</b> )	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 ( <b>0.05</b> )	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

individual models solely contain the weekday and the date as the number of days from the start of the study period.

Depending on the individual, predicting questionnaire answers with variables related to software repositories can achieve a marginal value of  $R^2$  up to 0.26. This is in harsh contrast to the general model in Table 5, where the marginal  $R^2$  value did not exceed 0.02.

Table 6 shows models for predicting hurry for three individuals, with the  $R^2$  value varying between 0.10 and 0.26. Comparing the general model and the individual models with respect to hurry reveals the following. Productivity is associated with reduced

**Table 9** Generalized linear mixed models predicting interruptions for the next day with today’s repository variables. A p-value of 0.05 or less is denoted in **bold**

Variable:	DevA	DevB	DevC	DevD
prod	0.06 (0.30)	-0.45 ( <b>0.02</b> )	-0.09 (0.42)	
nchat	< -0.01 (0.99)	-0.09 (0.60)	-0.06 (0.68)	
pval	-0.11 (0.28)	-0.19 (0.27)	-0.03 (0.88)	
nval	0.08 (0.28)	0.03 (0.90)	-0.03 (0.84)	
har	0.03 (0.77)	0.15 (0.39)	0.15 (0.44)	
lar	0.03 (0.60)	-0.12 (0.41)	-0.31 ( <b>0.04</b> )	
joyemo	-0.04 (0.25)	-0.01 (0.92)	0.01 (0.97)	
scsemo	0.15 (0.66)	-0.61 (0.56)	1.5 (0.17)	
meeting	0.14 (0.13)	-0.10 (0.68)	0.11 (0.45)	
failure event	-0.01 (0.83)	< -0.01 (0.99)	-0.23 (0.06)	
Marginal $R^2$	0.06	0.10	0.09	
Conditional $R^2$	0.06	0.10	0.09	
Null Model C. $R^2$	0	0	0	

**Table 10** Generalized linear mixed models predicting ineffective software development for the next day with today’s repository variables. A p-value of 0.05 or less is denoted in **bold**

Variable:	DevA	DevB	DevC	DevD
prod	0.02 (0.43)	−0.07 (0.64)	< −0.01 (0.99)	
nchat	<0.01 (0.95)	0.02 (0.86)	0.02 (0.85)	
pval	−0.01 (0.89)	−0.08 (0.54)	0.02 (0.87)	
nval	0.07 (0.06)	−0.09 (0.58)	−0.13 (0.26)	
har	−0.05 (0.32)	0.22 (0.12)	0.15 (0.30)	
lar	0.01 (0.85)	0.01 (0.92)	< −0.01 (0.98)	
joyemo	< −0.01 (0.90)	−0.03 (0.62)	−0.02 (0.83)	
scsemo	0.07 (0.70)	−0.05 (0.95)	−0.02 (0.97)	
meeting	0.05 (0.32)	−0.10 (0.63)	0.10 (0.38)	
failure event	0.01 (0.65)	−0.02 (0.63)	−0.11 (0.24)	
Marginal $R^2$	0.02	0.06	0.02	
Conditional $R^2$	0.02	0.06	0.02	
Null Model C. $R^2$	0	0	0	

hurry for developer B which it opposed the general model. Such opposing results between developers explain the low  $R^2$  value in the general model.

Table 7 has two models, as for two individuals the model did not converge. For the two developers, the marginal  $R^2$  values were at 0.10 and 0.09 respectively. For developer A, a p-value of 0.01 was calculated for productivity with a positive coefficient. For developer C, a negative coefficient and a p-value of 0.02 was calculated with a number of chat messages. Productivity also has a positive value for developer C and can also be found in the general model in Table 5 as a significant predictor. The number of chat messages was also negative for developer A, but it cannot be found in the general model. Other predictors that have the same sign for the two individuals are failure events and meetings.

**Table 11** Generalized linear mixed models predicting independence for the next day with today’s repository variables. A p-value of 0.05 or less is denoted in **bold**

Variable:	DevA	DevB	DevC	DevD
prod			−0.01 (0.94)	
nchat			−0.16 (0.16)	
pval			0.20 (0.23)	
nval			0.23 (0.08)	
har			0.12 (0.42)	
lar			−0.19 (0.11)	
joyemo			−0.02 (0.84)	
scsemo			−0.33 (0.70)	
meeting			0.14 (0.21)	
failure event			−0.09 (0.38)	
Marginal $R^2$			0.13	
Conditional $R^2$			0.13	
Null Model C. $R^2$			0	

Table 8 shows four models depicting the prediction of sleeping problems by the respondents. The marginal  $R^2$  values vary between 0.10 and 0.24. Significant predictors were high arousal for developer A with a positive relationship and p-value of 0.02. For developer Bm, the significant predictor was meetings with a negative relationship and a p-value of 0.05. Lastly, for developer C, the predictor was joy emoticons and emoji with a p-value of 0.04 and a negative relationship. None of these predictors were in the general model in Table 5. Uniform signs across the developers' could be found for negative valence in a negative relationship with sleeping problems.

Generally, individual models for the prediction of interruptions and ineffective software development achieve lower marginal  $R^2$  values compared to the other three questionnaire questions, with only one model achieving a marginal  $R^2$  value of 0.1. No statistically significant predictors could be found for ineffective software development, but two could be found for interruptions in Table 9. These are productivity for developer B with a negative relationship and a p-value of 0.02, and for developer C low arousal with a negative relationship and a p-value of 0.04. None of these was a significant predictor in the general model in Table 5. Lastly, Table 11 shows models for individuals predicting independence for one individual. The marginal  $R^2$  value is 0.13. There were no statistically significant predictors (Table 10).

Summary: We found no general model to predict software developer's well-being from software repositories. Yet, it seems that the well-being of each individual has different predictors.

### 4.3 RQ3—Can Software Developers' Well-Being and Actions Predict Software Developers' Productivity?

In RQ3, we examined whether developer productivity measured as a factor can be predicted with all of the other factors of our model, that is both the remaining software repository variables, as well as the questionnaire answers. The productivity factor consists of the number of commits, lines of code, and the number of files changed (Fig. 1).

Table 12 shows five different models for predicting productivity, one made using all the data and four made using individual developers with the most answers to the questionnaire, similar to RQ2. The  $R^2$  values for the fixed effects show that again, random effects explain more than fixed effects, with marginal  $R^2$  value of 0.03 and conditional  $R^2$  value of 0.52. Again, the random effects refer to control variables, which are used to explain the predicted variable, that is the respondent ID, the day of the week, and the date as a number from the start of the study, with the first day being one.

The three models showing individuals in Table 12 show individual variability, as only one predictor is statistically significant for one developer. Predictors with the same sign for all three individuals are the number of chat messages, negative valence, failure events, and independence. We can also see that the marginal  $R^2$  value rises from 0.03 of the general model to 0.05–0.20 depending on the individual.

This result is highly similar to what we observed in RQ2. To summarize, how experienced well-being and actions predict productivity significantly vary between individuals.

### 4.4 RQ4—Can Interviews Give Further Information About Experienced Well-Being of Software Developers?

**Motivation** We wanted to better understand the reason behind the numbers gathered with the daily questionnaire. With interviews, we also hoped to understand better what happened

**Table 12** 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**

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 ( <b>0.01</b> )	0.27 ( <b>&lt;0.001</b> )	-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	

when a particular event occurred. For example, whether meetings with the customer were attended by all developers, and what actions a developer had to do when tests for production failed. We also wanted to explore how instant messaging was used in the project, to possibly offer some explanations of our results. Finally, we asked questions related to emoticon and emoji usage, to better understand their usage and meaning in the project chat logs.

**Experience Sampling Procedure** All three interviewees, university graduates themselves, mentioned the primary motivation for answering the questionnaire was to offer helpful contributions to science. When asked of the possibility for minor rewards, such as movie tickets, developer 1 said: "I don't believe that those movie ticket thingies motivate working people. It is not about monetary compensation". Email messages were also described to be a good way of sharing the link to the questionnaire by all interviewees, as having one's email client open at work was described as being part of the job.

### Leadership Style, Company Culture, Way of Working with Respect to the Questionnaire

The project manager described their leadership style to be more facilitating and supportive. In practice, it meant that nobody was ever assigned to specific tasks, but that the developers chose their tasks from a list for the next sprint. The project manager expanded on this: "Probably a manager wouldn't fare long at the company, who would be saying you do this, you do this and so on". Both developers expressed that the project team had plenty of independence for making decisions and that the employer did not intervene in day to day decisions.

Furthermore, the project manager told us that the guidelines for developers were to commit small logical changes. Both of the developers backed this up in their interviews, and

perhaps as a result, neither of the developers recalled any bigger merge conflicts. We believe this is an important context worth mentioning for the models in the previous subsections.

**Hurry** Overall, both developers described the project as being without much time pressure. Developer 2 explained: “I would say that at a general level there was never a terrible hurry... I never felt like somebody was looking over my shoulder; that exactly this task should be ready by a deadline. I knew that if it wouldn’t be completed, nothing too terrible would happen.”. Furthermore, both developers described that, while hard deadlines did exist, the needed features were always ready well before this deadline. In the words of developer 1: “I never felt when we were going to production, that the project is going to cause so much hurry, but well, the version we had is already good enough.”.

Developer 1 offered this after-the-fact explanation: “Well, I believe, in this project, the feeling of hurry, has been precisely that you don’t have time to develop, but 70% of your time is going to everything else. When you are not in a hurry, you have seven and a half hours to code”. They also further explained “When I feel like I have to get something done... I don’t partake in internal educational events or other training, but I focus on developing the project. And otherwise, maybe I focus more on developing features rather than general project work”. Developer 1 also mentioned that writing tests is a part that could be easily skipped when feeling hurried: “... The feeling of hurry starts to come when I am implementing tests. But you still have to write the tests.”. The developer thus wrote the tests, but it felt like a part that could be skipped in a pinch.

**Role of Instant Messaging** All three interviewees agreed that project chat was used for communicating work and technical aspects the overwhelming majority of the time. Another company-wide chat mechanism exists for discussions related to free time, which was not part of our data sources. Two of the interviewees expanded that employees were urged to discuss technical aspects of work specifically on chat over and alongside face-to-face discussions. The benefits mentioned were traces to communications, coordination of expertise with everyone having the same access to information, better focus without interruptions in the shared working space (as opposed to face-to-face communication), and that the team would be aware of issues and solutions related to current events. One example topic for discussion would be for example, whether to integrate a specific new test automation tool to be part of the development process.

One negative consequence of the chat system was mentioned by a single respondent. Private messages from the chat system were seen as interrupting, as they felt there was a higher urgency to respond since a response was demanded specifically from them. This would be the case when the respondent was seen as an expert on some topic, and their opinion and expertise was valued and demanded by the person sending the private message. We want to note that our quantitative data does not include private messages.

According to the developers, some of the emoji used were quite specific to the context and were related to the humor in the project. For example, emojis related to parrots (e.g., “partyparrot”) were used when things went well or the developer felt something was accomplished. Emojis related to shoveling and a car jack were used when problems arose. The supporting element of the instant messaging channel in relation to the usage of emoji was highlighted by developer 1: “in those moments when you felt frustrated or irritated, then you would seek support with “in the trenches”-kind of humor”. We also note that we used this information on the classification of emoji for the quantitative analysis described above.

**Job Events** The project manager described the meetings with the customer in this project as “very long”, with meetings usually taking three hours. The meetings were “open meetings”. The project manager further explained that their goal was to circulate the developers to the meetings as they were needed based on their expertise related to the project. For example, when the topic would be a feature, only those who had developed the feature would be in attendance during that part of the meeting. However, the project manager was present from start to finish in the meetings. The project manager further elaborated on this: “Oh well, the developer does not want to sit in the meetings”.

Neither of the developers could recall situations in which they had to extensively prepare for the meetings. Both developers agreed that some preparation was needed, but it only required thinking about how to demonstrate and what to say about the features they had developed. Developer 2 described the preparation as solely consisting of looking at the agenda, and knowing which development branch in the version control system was the right one for the demonstration when needed. Developer 1 said that the continuous deployment eased the meetings: “The new code went to the customers’ environment, so they could go and use it. I never needed Powerpoint presentations”.

The project manager had the poorest recollections about whether production tests had failed, as significant problems related to hosting the service arose during our study period. However, neither of the developers shared these recollections, perhaps in part because, for hosting and optimization related issues, extra personnel from the operation team outside of the normal development team were involved. While an instant reaction was demanded from the developers, neither of them saw these as particularly bad. Developer 1 explicated: “It was never a catastrophe, as it only meant that updates to the staging environment would stop and production would not be updated the next morning. Those whose code changes broke the build usually started to fix it as soon as possible. Usually, it was not a big deal.”.

## 5 Discussion

Ultimately, the main finding of our study is that predicting well-being strongly depends on the individual. While the marginal  $R^2$  value did not rise above 0.26 in the models of the individual, such lower  $R^2$  values have been reported in more technical studies as well. For example, depending on the project studied, bug prediction models have achieved  $R^2$  values in the 0.20’s (Giger et al. (2011) and D’Ambros et al. (2010)). Is our study a negative result? On the general level, it is as we cannot find shared predictors that would work on all individuals. But on the individual level, it is not as individual predictors were in line with some past work.

We cannot establish strong links between repository variables and our questionnaire variables related to well-being. We also do not see the links we had between the questionnaire and software repository variables with logistic regression in our prior work with the same dataset (Kuuttila et al. 2018b). We think this is mostly because of the additional control variables we used as random effects in our model. The main random effect explaining the majority of the variation is the respondent ID in the generalized linear mixed effects models. Our general models for prediction shown in Sections 4.2 and 4.3 would look much closer to our previous work if we had not controlled for the individual.

Additionally, repository data are inherently incomplete. However, in some ways repository data will always be incomplete, as Aranda and Venolia (2009) have noted: “the histories of even simple bugs are strongly dependent on social, organizational, and technical

knowledge that cannot be solely extracted through automation of electronic repositories, and that such automation provides incomplete and often erroneous accounts of coordination.”. Therefore, repositories always reflect only part of software engineering work actions. Furthermore, events outside work will influence how people feel and sleep, which can influence the questionnaire answers.

Our results are in line with some previous negative results on sentiment analysis studies, e.g. Jongeling et al. (2017) and Lin et al. (2018). Even under laboratory conditions, valence explained 27%, and arousal 0.5% of perceived progress in the software development task (Girardi et al. 2020), which is comparable to our productivity measure and the models made with individual data in Table 12. However, we did not find a link between positive valence measured from the chat system and our measured productivity.

In general, the interviews demonstrated that no big deadline pressure or prolonged time pressures were felt during the project, though variance among the answers during the project can be seen in Figs. 1 and 2 in our prior work (Kuuttila et al. 2018a). Observing distress and time pressure could be easier when they are more frequent in the software project. Software projects having less time pressure using agile methods are also in line with results from our prior literature review (Kuuttila et al. 2020a).

We also observe that sending more messages to instant messaging chat was not tied to any clear negative effects. This finding is contrary to some previous work in the information technology field (Cameron and Webster 2005; Sykes 2011) where instant messaging was linked to more negative outcomes. The link to more interruptions reported by Sykes (2011) was also reported by one developer during our interviews, but only when using private messages rather than the project-wide chat. Based on the evidence gathered in this study, we believe that using instant messaging applications during software development projects can be beneficial if it is used as a collaborative tool to coordinate expertise, rather than for delivering commands or checking up on whether someone is working. A more facilitating leadership style and a company culture that allowed more independent decisions seemed to be a key contextual difference in this project compared to prior studies.

While the sentiment analysis we performed is quite rudimentary, we demonstrated some links between well-being and variables related to sentiment and emoticon usage. In Table 5 positive valence has a positive coefficient with independence. Moreover, one novel aspect of our work is the usage of emoticons and emoji, in Table 5 emoticons and emoji related to sadness, confusion, and surprise were statistically significant predictors with regards to independence and hurry.

Finally, we think that one point raised in the interviews is interesting and could be considered in future experience sampling studies. One of the developers mentioned feeling hurried especially when they did not have time for programming and had to do tasks other than development. Such tasks could be related to design, job training and quality assurance. Depending on the project context, one question in a future questionnaire could ask how the developer divided their time between different tasks.

## 6 Threats to Validity

### 6.1 Internal Validity

The interviews were conducted a considerable amount of time after the questionnaire, and partly because of this we could not interview all the developers who answered the questionnaire. However, the ones interviewed are also some of the ones with the highest



response rates to the questionnaire. We tried to help remembrance by sending individual level graphs of the questionnaire answers to the interviewees. We also quantified the interviewees answers, to see how uniform the answers to questions were. Time of the week and month when answering also can influence the answer, which we tried to control with variables in the generalized mixed effects linear models. Other individual traits such as seniority and gender can have an effect, but due to anonymity issues and a low sample size, we do not report these. Experiences and events not related to work can also influence well-being. Thus confounding variables can have an effect on our mixed-effects models.

With regards to generalized linear mixed models, Bolker and et al (2020) collected an encompassing discussion on how to decide whether a variable is fixed or random for generalized linear mixed effects models. Crawley (2002) advocated using variables as fixed effects when there not enough levels inside random effects, and Bolker and et al. (2020) further sees six levels inside a random effect as the absolute minimum. Thus the levels inside random effects (weekday, respondent ID) can have an effect on our models.

The complexity of random effects structures together with sample size influence model convergence Barr et al. (2013). Indeed we did have some convergence issues specifically when producing models for individuals where the sample size is lower than the general model. In our case, we simplified the random effects structure by using different moving averages for auto-correlation that helped to get rid of some convergence issues.

## 6.2 External Validity

The questionnaire was only administered at a single software company with a single software project. This diminishes the generalizability of our results. We tried to contextualize our study partly with the interviews performed in Section 4.4. Major context factors include the company culture, which was described as facilitating and allowing independence for developers, and moreover, without major time pressures. Other contextual factors include an agile way of working, pushing code to production daily, as well as having no big interrogations. We believe our results would be replicable in such a context. However, our study is just one project in a single company, in a single country, and hence, how these different contexts alter the results is yet to be discovered.

## 6.3 Construct Validity

The sentiment analysis we performed is rudimentary, mainly because the development team used the Finnish language for instant messages. This severely limited the choice of sentiment analysis tools we could use for this study. The valence lexicon used is not widely known. However, we decided to use it because it is developed specifically for the software engineering context. Studying company-specific jargon would improve the validity of the constructs produced by sentiment analysis, but doing it on a large scale would be a study on its own. We did take some information about the emoticons used into account acquired in the interviews.

Debate on the usage of single-item measures in experience sampling studies exists. Specifically, Rossiter (2002) argues for the validity of “doubly concrete” constructs in single-item measurements, that is constructs for which the object and attribute of measurement are unambiguous and clear for the raters. Evidence supporting this view is also presented by multitude of other studies, e.g. Bergkvist and Rossiter (2009) and Wanous et al. (1997). More discussion on the subject, including both supporting and contradictory evidence, can be found in an article by Fisher and To (2012). Based on the evidence, Fisher and

To (2012) see single-item measurements more valid when they are “straight forward unidimensional constructs in terms of current or very recent experience”, rather than complicated constructs that are rated retrospectively over a longer time span.

## 7 Conclusions

To our knowledge, we present a highly novel study. We observe software developers’ well-being with experience sampling over a period of eight months. Additionally, we explore the relationship between well-being and metrics mined from software repositories. If a strong link between well-being and software repositories could be established, this would mean that automated well-being monitoring of software developers would be possible.

Our results show that developers’ well-being varied individually rather than in a collective manner. We found that software engineering actions (fixed effects) mined mainly from software repositories are not good general predictors of well-being or productivity. Rather it is the individual (modeled as a random effect) that explains differences in well-being and productivity. We further investigated the individuals and found that models of well-being and productivity developed per individual performed better than general models. For example, the top general model had a marginal  $R^2$  value of 0.02 while in the individual models top marginal  $R^2$  value was 0.26. Thus, adage about predicting “some of the people some of the time” holds (Bem and Allen 1974).

Future studies on this topic should be improved. A higher number of respondents should be used. However, convincing larger groups to respond to daily surveys over periods of several months is likely to be challenging. Perhaps, the time duration for the survey responses could be shorter, e.g. a month, if the number of individuals responding could be increased to tens of developers. With the increased number of individuals, one could meaningfully study if the individual differences in well-being and productivity that we observed are due to different roles, e.g. senior versus junior developers could have different well-being predictors in software repositories. If one could collect responses from hundreds of developers, then perhaps even personality types could be taken into account (Eysenck et al. 2020).

Future studies in software engineering using experience sampling also offer interesting possibilities. Experience sampling can be used to study a multitude of factors related to software engineering. These include the effects of different kinds of processes, techniques, and ways related to software development work, such as the adoption of agile, teleworking, resistance to change, and organizational justice. We also believe that replicating well-being studies in different software development contexts is beneficial, to better understanding contextual factors.

**Acknowledgements** The first, second and third author have been supported by Academy of Finland grant 298020. The first author has been supported by Kaute-foundation.

**Funding** Open access funding provided by University of Oulu including Oulu University Hospital.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Adams WC (2015) Conducting semi-structured interviews. *Handbook of Practical Program Evaluation*
- Åkerstedt T, Knutsson A, Westerholm P, Theorell T, Alfredsson L, Kecklund G (2002) Sleep disturbances, work stress and work hours: a cross-sectional study. *J Psychosom Res* 53(3):741–748
- Alliger GM, Williams KJ (1993) Using signal-contingent experience sampling methodology to study work in the field: a discussion and illustration examining task perceptions and mood. *Pers Psychol* 46(3):525–549
- Anderson JC, Gerbing DW (1984) The effect of sampling error on convergence, improper solutions, and goodness-of-fit indices for maximum likelihood confirmatory factor analysis. *Psychometrika* 49(2):155–173
- Aranda J, Venolia G (2009) The secret life of bugs: going past the errors and omissions in software repositories. In: 2009 IEEE 31st international conference on software engineering. IEEE, pp 298–308
- Bakker AB, Demerouti E (2007) The job demands-resources model: state of the art. *J Manag Psychol* 22(3):309–328
- Bakker AB, Demerouti E, Euwema MC (2005) Job resources buffer the impact of job demands on burnout. *J Occup Health Psychol* 10(2):170
- Barr DJ, Levy R, Scheepers C, Tily HJ (2013) Random effects structure for confirmatory hypothesis testing: keep it maximal. *J Mem Lang* 68(3):255–278
- Barton K (2009) Mumin: multi-model inference. <http://r-forger-project.org/projects/mumin/>
- Bem DJ, Allen A (1974) On predicting some of the people some of the time: the search for cross-situational consistencies in behavior. *Psychol Rev* 81(6):506
- Bergkvist L, Rossiter JR (2009) Tailor-made single-item measures of doubly concrete constructs. *Int J Advert* 28(4):607–621
- Bjarnason E, Hess A, Svensson RB, Regnell B, Doerr J (2014) Reflecting on evidence-based timelines. *IEEE Softw* 31(4):37–43
- Boehm BW et al (1981) *Software engineering economics*, vol 197. Prentice-Hall, Englewood Cliffs
- Bolker B et al (2020) Glmm faq. <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#should-i-treat-factor-xxx-as-fixed-or-random>
- Bollen K (2001) Indicator: methodology. In: Smelser NJ, Baltes PB (eds) *International encyclopedia of the social & behavioral sciences*. Pergamon, Oxford, pp 7282–7287. <https://doi.org/10.1016/B0-08-043076-7/00709-9>. <http://www.sciencedirect.com/science/article/pii/B0080430767007099>
- Bruckhaus T, Madhavii N, Janssen I, Henshaw J (1996) The impact of tools on software productivity. *IEEE Softw* 13(5):29–38
- Brumby DP, Janssen CP, Mark G (2019) How do interruptions affect productivity? In: *Rethinking productivity in software engineering*. Springer, pp 85–107
- Cameron AF, Webster J (2005) Unintended consequences of emerging communication technologies: Instant messaging in the workplace. *Comput Hum Behav* 21(1):85–103
- Chrousos GP, Gold PW (1992) The concepts of stress and stress system disorders: overview of physical and behavioral homeostasis. *JAMA: J Am Med Assoc* 267(9):1244–1252
- Claes M, Mäntylä M, Farooq U (2018a) On the use of emoticons in open source software development. In: *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*. ACM, p 50
- Claes M, Mäntylä M, Kuutila M, Adams B (2018b) Do programmers work at night or during the weekend? In: 2018 IEEE/ACM 40th international conference on software engineering (ICSE). IEEE, pp 705–715
- Crawley MJ (2002) *Statistical computing an introduction to data analysis using S-Plus*. 001.6424 C73
- D'Ambros M, Lanza M, Robbes R (2010) An extensive comparison of bug prediction approaches. In: 2010 7th IEEE working conference on mining software repositories (MSR 2010). IEEE, pp 31–41
- De Kloet ER, Joëls M, Holsboer F (2005) Stress and the brain: from adaptation to disease. *Nat Rev Neurosci* 6(6):463
- Demerouti E, Bakker AB, Nachreiner F, Schaufeli WB (2001) The job demands-resources model of burnout. *J Appl Psychol* 86(3):499
- Diaz M, Sligo J (1997) How software process improvement helped motorola. *IEEE Softw* 14(5):75–81
- Dickersin K (1990) The existence of publication bias and risk factors for its occurrence. *JAMA* 263(10):1385–1389
- Diener E, Suh EM, Lucas RE, Smith HL (1999) Subjective well-being: three decades of progress. *Psychol Bull* 125(2):276
- Dirnagl U, Lauritzen M (2010) Fighting publication bias: introducing the negative results section

- Donald I, Taylor P, Johnson S, Cooper C, Cartwright S, Robertson S (2005) Work environments, stress, and productivity: an examination using asset. *Int J Stress Manag* 12(4):409
- Dueñas S, Cosentino V, Robles G, Gonzalez-Barahona JM (2018) Perceval: software project data at your will. In: Proceedings of the 40th international conference on software engineering: companion proceedings. ACM, pp 1–4
- Elo AL, Leppänen A, Jahkola A (2003) Validity of a single-item measure of stress symptoms. *Scand J Work Environ Health* 29(6):444–451
- Elovainio M, Heponiemi T, Jokela M, Hakulinen C, Pesseau J, Aalto AM, Kivimäki M (2015) Stressful work environment and wellbeing: what comes first? *J Occup Health Psychol* 20(3):289
- Eysenck SB, Barrett PT, Saklofske DH (2020) The junior Eysenck personality questionnaire. *Personality and Individual Differences*, p 109974
- Fanelli D (2012) Negative results are disappearing from most disciplines and countries. *Scientometrics* 90(3):891–904
- Fisher CD, To ML (2012) Using experience sampling methodology in organizational behavior. *J Organ Behav* 33(7):865–877
- 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 Trans Softw Eng* 16:1–19
- Giger E, Pinzger M, Gall HC (2011) Comparing fine-grained source code changes and code churn for bug prediction. In: Proceedings of the 8th working conference on mining software repositories, pp 83–92
- Girardi D, Novielli N, Fucci D, Lanubile F (2020) Recognizing developers' emotions while programming. In: Proceedings of the ACM/IEEE 42nd international conference on software engineering, pp 666–677
- Goldberg DP, 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. ACM, pp 25–32
- Harris PE (1989) The nurse stress index. *Work Stress* 3(4):335–346
- Hassan AE (2009) Predicting faults using the complexity of code changes. In: Proceedings of the 31st international conference on software engineering. IEEE Computer Society, pp 78–88
- Hove SE, Anda B (2005) Experiences from conducting semi-structured interviews in empirical software engineering research. In: 11th IEEE international software metrics symposium (METRICS'05). IEEE, 10 pp
- Hyndman RJ, Khandakar Y et al (2007) Automatic time series for forecasting: the forecast package for R 6/07. Monash University, Department of Econometrics and Business Statistics
- Ilies R, Judge TA (2004) An experience-sampling measure of job satisfaction and its relationships with affectivity, mood at work, job beliefs, and general job satisfaction. *Eur J Work Organ Psychol* 13(3):367–389
- Jenkins CD, Stanton BA, Niemcryk SJ, Rose RM (1988) A scale for the estimation of sleep problems in clinical research. *J Clin Epidemiol* 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). IEEE, pp 531–535
- Jongeling R, Sarkar P, Datta S, Serebrenik A (2017) On negative results when using sentiment analysis tools for software engineering research. *Empir Softw Eng* 22(5):2543–2584
- Karasek R (1990) Healthy work. Stress, productivity, and the reconstruction of working life
- Karasek R, Brisson C, Kawakami N, Houtman I, Bongers P, Amick B (1998) The job content questionnaire (jqc): an instrument for internationally comparative assessments of psychosocial job characteristics. *J Occup Health Psychol* 3(4):322
- Kimhy D, Delespaul P, Corcoran C, Ahn H, Yale S, Malaspina D (2006) Computerized experience sampling method (esmc): assessing feasibility and validity among individuals with schizophrenia. *J Psychiatr Res* 40(3):221–230
- Krippendorff K (1980) Reliability. Wiley Online Library
- Krippendorff K (2011) Computing krippendorff's alpha-reliability. [https://repository.upenn.edu/asc\\_papers/43/](https://repository.upenn.edu/asc_papers/43/)
- Kross E, Verduyn P, Demiralp E, Park J, Lee DS, Lin N, Shablack H, Jonides J, Ybarra O (2013) Facebook use predicts declines in subjective well-being in young adults. *PLoS One* 8(8):e69841
- Kuutila M, Mäntylä MV, Claes M, Elovainio M (2018a) Daily questionnaire to assess self-reported well-being during a software development project. In: Proceedings of the 3rd international workshop on emotion awareness in software engineering. ACM, pp 39–43

- Kuutila M, Mäntylä MV, Claes M, Elovainio M, Adams B (2018b) Using experience sampling to link software repositories with emotions and work well-being. In: Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement. ACM, p 29
- Kuutila M, Mäntylä M, Farooq U, Maëlick C (2020a) Time pressure in software engineering: a systematic review. *Inf Softw Technol* 121:106257. <https://doi.org/10.1016/j.infsof.2020.106257>. <http://www.sciencedirect.com/science/article/pii/S0950584920300045>
- Kuutila M, Mäntylä MV, Claes M (2020b) Chat activity is a better predictor than chat sentiment on software developers productivity. In: Proceedings of the IEEE/ACM 42nd international conference on software engineering workshops, pp 553–556
- Lenberg P, Feldt R, Wallgren LG (2015) Behavioral software engineering: a definition and systematic literature review. *J Syst Softw* 107:15–37
- 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
- Linton SJ (2004) Does work stress predict insomnia? A prospective study. *Br J Health Psychol* 9(2):127–136
- Liu B (2009) Handbook chapter: sentiment analysis and subjectivity. *Handbook of natural language processing*. Handbook of Natural Language Processing Marcel Dekker Inc, New York
- Mäntylä MV, Petersen K, Lehtinen TO, 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. ACM, pp 83–94
- Mäntylä MV, 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). IEEE, pp 198–202
- Mark G, Gudith D, Klocke U (2008) The cost of interrupted work: more speed and stress. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 107–110
- Marsh HW, Balla JR, Hau KT (1996) An evaluation of incremental fit indices: a clarification of mathematical and empirical properties. *Advanced structural equation modeling: Issues and techniques*, pp 315–353
- McEwen BS, Sapolsky RM (1995) Stress and cognitive function. *Curr Opin Neurobiol* 5(2):205–216
- Miller HJ, 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
- Miner A, Glomb T, Hulin C (2005) Experience sampling mood and its correlates at work. *J Occup Organ Psychol* 78(2):171–193
- Mockus A, Fielding RT, Herbsleb JD (2002) Two case studies of open source software development: Apache and Mozilla. *ACM Trans Softw Eng Methodol (TOSEM)* 11(3):309–346
- Murphy-Hill E, Jaspan C, Sadowski C, Shepherd D, Phillips M, Winter C, Knight A, Smith E, Jorde M (2019) What predicts software developers’ productivity? *IEEE Trans Softw Eng* 47(3):582–594
- Muthén LK, Muthén BO (2017) Unusual tli values. <https://www.statmodel.com/download/TLI.pdf>
- Nagy MS (2002) Using a single-item approach to measure facet job satisfaction. *J Occup Organ Psychol* 75(1):77–86
- Nakagawa S, Schielzeth H (2013) A general and simple method for obtaining  $r^2$  from generalized linear mixed-effects models. *Methods Ecol Evol* 4(2):133–142
- Nan N, Harter DE (2009) Impact of budget and schedule pressure on software development cycle time and effort. *IEEE Trans Softw Eng* 35(5):624–637
- 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). IEEE, pp 364–375
- Pinheiro J, Bates D, DebRoy S, Sarkar D, Heisterkamp S, Van Willigen B, Maintainer R (2017) Package ‘nlme’. *Linear and nonlinear mixed effects models*, version 3(1)
- Pitkänen H (2012) Voikko—free linguistic software and data for finnish. <https://voikko.puimula.org/>
- Plutchik R (1991) *The emotions*. University Press of America
- Rossiter JR (2002) The c-oar-se procedure for scale development in marketing. *Int J Res Market* 19(4):305–335
- Russell JA (2003) Core affect and the psychological construction of emotion. *Psychol Rev* 110(1):145
- Schermelleh-Engel K, Moosbrugger H, Müller H et al (2003) Evaluating the fit of structural equation models: tests of significance and descriptive goodness-of-fit measures. *Methods Psychol Res Online* 8(2):23–74
- Schmidt C (2004) The analysis of semi-structured interviews. A companion to qualitative research, pp 253–258

- Schuler RS (1980) Definition and conceptualization of stress in organizations. *Organ Behav Hum Perform* 25(2):184–215
- Schulte P, Vainio H (2010) Well-being at work—overview and perspective. *Scand J Work Environ Health* 36(5):422–429
- Schwarz N, Clore GL (1983) Mood, misattribution, and judgments of well-being: informative and directive functions of affective states. *J Pers Social Psychol* 45(3):513
- Scollon CN, Prieto CK, Diener E (2009) Experience sampling: promises and pitfalls, strength and weaknesses. In: *Assessing well-being*. Springer, pp 157–180
- Semmer N, Zapf D, Greif S (1996) Shared job strain: a new approach for assessing the validity of job stress measurements. *J Occup Organ Psychol* 69(3):293–310
- Silla I, Gamero N (2014) Shared time pressure at work and its health-related outcomes: job satisfaction as a mediator. *Eur J Work Organ Psychol* 23(3):405–418
- Singh P, Suar D (2013) Health consequences and buffers of job burnout among Indian software developers. *Psychol Stud* 58(1):20–32
- Snir R, Zohar D (2008) Workaholism as discretionary time investment at work: an experience-sampling study. *Appl Psychol* 57(1):109–127
- Sonnentag S, Brodbeck FC, Heinbokel T, Stolte W (1994) Stressor-burnout relationship in software development teams. *J Occup Organ Psychol* 67(4):327–341
- Sun J (2005) Assessing goodness of fit in confirmatory factor analysis. *Meas Eval Counsel Dev* 37(4):240–256
- Svenson O (1993) *Time pressure and stress in human judgment and decision making*. Springer Science & Business Media
- Sykes ER (2011) Interruptions in the workplace: a case study to reduce their effects. *Int J Inf Manag* 31(4):385–394
- Taipale S, Selander K, Anttila T, Nätti J (2011) Work engagement in eight European countries: the role of job demands, autonomy, and social support. *Int J Sociol Social Policy* 31(7/8):486–504
- Tarafdar M, Tu Q, Ragu-Nathan BS, Ragu-Nathan T (2007) The impact of technostress on role stress and productivity. *J Manag Inf Syst* 24(1):301–328
- Thompson B (2004) *Exploratory and confirmatory factor analysis*. American Psychological Association
- Tregubov A, Boehm B, Rodchenko N, Lane JA (2017) Impact of task switching and work interruptions on software development processes. In: *Proceedings of the 2017 international conference on software and system process*. ACM, pp 134–138
- Tucker LR, Lewis C (1973) A reliability coefficient for maximum likelihood factor analysis. *Psychometrika* 38(1):1–10
- Vrijkotte TG, Van Doornen LJ, De Geus EJ (2000) Effects of work stress on ambulatory blood pressure, heart rate, and heart rate variability. *Hypertension* 35(4):880–886
- Wagner S, Ruhe M (2018) A systematic review of productivity factors in software development. [arXiv:180106475](https://arxiv.org/abs/180106475)
- Wanous JP, Reichers AE, Hudy MJ (1997) Overall job satisfaction: how good are single-item measures? *J Appl Psychol* 82(2):247
- West SG, Hepworth JT (1991) Statistical issues in the study of temporal data: daily experiences. *J Personal* 59(3):609–662
- Xanthopoulou D, Bakker AB, Dollard MF, Demerouti E, Schaufeli WB, Taris TW, Schreurs PJ (2007) When do job demands particularly predict burnout? the moderating role of job resources. *J Manag Psychol* 22(8):766–786
- Xanthopoulou D, Bakker AB, Demerouti E, Schaufeli WB (2009) Reciprocal relationships between job resources, personal resources, and work engagement. *J Vocat Behav* 74(3):235–244

## Affiliations

Miikka Kuutila<sup>1</sup>  · Mika Mäntylä<sup>1</sup> · Maëlick Claes<sup>1</sup> · Marko Elovainio<sup>2</sup> · Bram Adams<sup>3</sup>

Mika Mäntylä  
mika.mantyla@oulu.fi

Maëlick Claes  
maelick.claes@oulu.fi

Marko Elovainio  
marko.elovainio@helsinki.fi

Bram Adams  
bram.adams@queensu.ca

<sup>1</sup> M3S, ITEE, University of Oulu, Oulu, Finland

<sup>2</sup> Department of Psychology and Logopedics, University of Helsinki, Helsinki, Finland

<sup>3</sup> School of Computing, Queen's University, Kingston, Ontario, Canada