arXiv:1703.04993v3 [cs.SE] 10 May 2017

On the Unhappiness of Software Developers

Daniel Graziotin
Institute of Software Technology
University of Stuttgart
Stuttgart, Germany
daniel.graziotin@informatik.uni-stuttgart.de

Fabian Fagerholm
Department of Computer Science
University of Helsinki
Helsinki, Finland
fabian.fagerholm@helsinki.fi

Xiaofeng Wang
Faculty of Computer Science
Free University of Bozen-Bolzano
Bolzano-Bozen, Italy
xiaofeng.wang@unibz.it

Pekka Abrahamsson
Department of Computer and Information Science (IDI)
NTNU
Trondheim, Norway
pekkaa@ntnu.no

ABSTRACT

The happy-productive worker thesis states that happy workers are more productive. Recent research in software engineering supports the thesis, and the ideal of flourishing happiness among software developers is often expressed among industry practitioners. However, the literature suggests that a cost-effective way to foster happiness and productivity among workers could be to limit unhappiness. Psychological disorders such as job burnout and anxiety could also be reduced by limiting the negative experiences of software developers. Simultaneously, a baseline assessment of (un)happiness and knowledge about how developers experience it are missing. In this paper, we broaden the understanding of unhappiness among software developers in terms of (1) the software developer population distribution of (un)happiness, and (2) the causes of unhappiness while developing software. We conducted a large-scale quantitative and qualitative survey, incorporating a psychometrically validated instrument for measuring (un)happiness, with 2 220 developers, yielding a rich and balanced sample of 1 318 complete responses. Our results indicate that software developers are a slightly happy population, but the need for limiting the unhappiness of developers remains. We also identified 219 factors representing causes of unhappiness while developing software. Our results, which are available as open data, can act as guidelines for practitioners in management positions and developers in general for fostering happiness on the job. We suggest considering happiness in future studies of both human and technical aspects in software engineering.

CCS CONCEPTS

•**Social and professional topics → Project and people management;** •**Applied computing → Psychology;** •**Software and its engineering** → *Software creation and management; Programming teams;*

KEYWORDS

behavioral software engineering; developer experience; human aspects; affect; emotion; mood; happiness

1 INTRODUCTION

The need and importance of managing the individuals forming the software development workforce were identified early in software engineering research [41]. Management and people related challenges grow as the numbers of software companies and developers increase with the digitalization of existing businesses and startups founded on software from day one [59]. A practice that has emerged recently is to promote flourishing happiness among workers in order to enact the happy-productive worker thesis [72]. Notable Silicon Valley companies and influential startups are well known for their perks to developers [52]. Recognizing, managing, and improving the happiness of all stakeholders involved in producing software is essential to software company success [10].

A novel line of research belonging to behavioral software engineering [47] is emerging, focusing on the relationship between the happiness of developers and work-related constructs such as performance and productivity [19, 20, 32, 33, 35, 54, 56], and software quality [11, 44]. The empirical evidence indicates that happy developers perform better than unhappy developers [34]. The studies so far, including those by the present authors, imply that managers and team leaders should attempt to foster developer happiness.

There is the other side of the coin, though. Diener [12] and Kahneman [43] have suggested that objective happiness[1] can be assessed by the difference between experienced positive affect and experienced negative affect. The happiness equation suggests that maximizing happiness can be achieved by maximizing positive

---

[1]We are using the more colloquial term *happiness* instead of *subjective well-being* throughout the paper as it has historical meaning to research in organizational behavior and psychology [24]. Furthermore, as our view of *un*happiness contemplates it as the negative component of happiness, we interchange the two terms when dealing with quantifications of developers' (un)happiness.

experiences of individuals, minimizing their negative experiences, or both.

Software developers are prone to share horror stories about their working experience on a daily basis [34]. Those in managerial positions should attempt to understand the nature and dynamics of unhappiness in the workplace to create programs for preventing dysfunctional responses among employees [68]. Psychological disorders such as stress and burnout could be reduced by analyzing the negative affective experiences of developers and turning them positive [51]. Furthermore, the voice of practitioners should be heard in software engineering research – software developers want their unhappiness to be voiced out [22, 34]. For the previously stated reasons, there are calls to understand the benefits of limiting negative experiences on the job [3, 12, 24].

The current research on software developers' affective experience lacks a baseline estimation of the distribution of happiness among software developers, as well an understanding of the causes of unhappiness that would be based on a broad sample.

In this paper, we echo the previous calls and aim to *broaden the understanding of unhappiness among software developers*. Based on the existing literature, we set the following research questions (RQs).

RQ1 *What is the distribution of (un)happiness among software developers?*

RQ2 *What are the experienced causes for unhappiness among software developers while developing software?*

To answer the RQs, we conducted a large-scale quantitative and qualitative survey of 2 220 software developers in which we asked them to voice out causes of happiness as well as unhappiness. We computed the population estimate of happiness, found 219 causes of unhappiness, and showed that the most prevalent causes of unhappiness are external to developers, suggesting that managers and team leaders have a realistic chance of influencing the happiness of software developers at work. We archived the list of causes as open data [31].

2 BACKGROUND AND RELATED WORK

What is happiness, and what does it mean to be happy or unhappy? Intuitively, we could associate this question to the sensing of an individual's affect. We begin by discussing affect, emotions, and moods.

Russell [61] has provided a widely agreed definition of *affect* as "*a neurophysiological state that is consciously accessible as a simple, non-reflective feeling that is an integral blend of hedonic (pleasure– displeasure) and arousal (sleepy–activated) values*" (p. 147). That is, affect is how we feel at any given point in time, about anything, and this feeling is expressed in how pleasant and activated our state of mind is. We have argued elsewhere [37] that there are several theories and definitions for emotions and moods. For clarity and brevity, we use Russell's [61] theory for the present paper, which considers affect as the atomic unit upon which moods and emotions can be constructed. We consider *moods* as prolonged, unattributed affect, and we consider *emotions* as interrelated events concerning a psychological object, i.e., an episodic process of perception of affect that is clearly bounded in time, in line with several other authors, e.g., [21, 44].

From a hedonistic point of view, a blend of affect constitutes an individual's *happiness* [39]. Happiness is a sequence of experiential episodes [39] and being happy (unhappy) corresponds with the frequency of positive (negative) experiences [50][2]. Frequent positive (negative) episodes lead to feeling frequent positive (negative) affect, which in turn leads to happiness (unhappiness), represented by a positive (negative) *affect balance* [13]. In brief, unhappy individuals are those who experience negative affect more often than positive affect, which is a condition that can be detected by a negative affect balance [13, 50].

2.1 Scale of Positive and Negative Experience

Recent studies have found several shortcomings in the Positive and Negative Affect Schedule (PANAS) [69], the most prominent measurement instrument for assessing happiness, in terms of its affect coverage [13, 49, 66] and neglect of cultural differences [49, 67]. New scales have been developed that attempt to address PANAS' limitations. Diener et al. [13] have presented the Scale of Positive and Negative Experience (SPANE), a short scale that assesses the happiness of participants by asking them to report the frequency of their positive and negative experiences during the last four weeks. SPANE has been reported to be capable of measuring positive and negative affect (and happiness) regardless of the sources, mental activation level, or cultural context, and it captures affect from the entire affective spectrum [13, 49]. Respondents are asked to report on their affect, expressed with adjectives that individuals recognize as describing emotions or moods, from the past four weeks in order to provide a balance between the sampling adequacy of affect and the accuracy of human memory to recall experiences [49], as well as to decrease the ambiguity of people's understanding of the scale itself [13].

SPANE has been validated to converge to other similar measurement instruments, including PANAS [13]. The scale provides good psychometric properties (validity and reliability) which were empirically demonstrated in several large-scale studies [6, 13, 16, 42, 49, 63, 64]. The scale has been proven consistent across full-time workers and students [63]. For these reasons (and for its brevity), we chose SPANE for the purpose of our research.

SPANE is a 12-item scale divided into two sub-scales of positive (SPANE-P) and negative (SPANE-N) experiences. The answers to the 12 items are given on a five-point scale ranging from 1 (*very rarely or never*) to 5 (*very often or always*). The SPANE-P and SPANE-N measures are the sum of the scores given to their respective six items, each ranging from 6 to 30. The two scores are further combined by subtracting SPANE-N from SPANE-P, resulting in the *Affect Balance (SPANE-B)* score. SPANE-B is an indicator of the happiness caused by how often positive and negative affect has been felt by a participant. SPANE-B ranges from −24 (*completely negative*) to +24 (*completely positive*).

2.2 Related Studies

Interest in studying the affect of software developers has risen considerably in the last five years, although we have just started

---

[2]Alternative views of happiness exist, e.g., the Aristotelian eudaimonia considers a person happy because (s)he conducts a satisfactory life full of quality [40]. We have also discussed the role of the centrality of affect in [35]. Current research in psychology supports the affect balance model as a valid approach to quantify happiness

to understand the tip of the iceberg [53]. To our knowledge, no studies have offered an estimation of the happiness distribution of developers, and only a small number of causes of developers' affect have been examined in a few studies. The present study addresses this research gap.

There are some initial indicators regarding developers' happiness. Generally speaking, studies indicate a positive relationship between the happiness of developers and their performance. Graziotin et al. [33] performed a quasi-experiment on the impact of affect on analytic problem solving and creative performance. The study itself was about consequences of happiness, thus not particularly related to the present article. Yet, Graziotin et al. observed that the sample distribution of happiness, measured using SPANE, was significantly greater than 0 (SPANE-B mean=7.58, 95% CI [5.29, 9.85]; median=9). The authors noted that the SPANE-B distribution did not resemble a normal distribution. However, the sample was very limited (N=42 BSc and MSc students of the same CS faculty); further exploration and validation were suggested based on the observations. We build on these initial observations in the present study.

Some studies have attempted to uncover issues related to affect using both qualitative and quantitative approaches with different degrees of automation. De Choudhury and Counts [9] investigated the expression of affect through the analysis of 204 000 micro-blogging posts from 22 000 unique users of a Fortune 500 software corporation. The sentiment analysis revealed that IT-related issues were often sources of frustration. Day-to-day demands, e.g., meetings, were also associated with negative affect.

Ford and Parnin [22] have explored frustration in software engineering through practitioner interviews. 67% of the 45 participants reported that frustration is a severe issue for them. As for the causes for such frustration, the authors categorized the responses as follows: not having a good mental model of the code (for the category "mapping behavior to cause"), learning curves of programming tools, too large task size, time required for adjusting to new projects, unavailability of resources (e.g., documentation, server availability, …), perceived lack of programming experience, not fulfilling the estimated effort for perceived simple problems, fear of failure, internal hurdles and personal issues, limited time, and issues with peers.

Graziotin et al. [35] conducted a qualitative study for constructing an explanatory process theory of the impact of affect on development performance. The theory was constructed by coding data coming from interviews, communications, and observations of two software developers working on the same project for a period of 1.5 months. The theory was built upon the concepts of events, affect, focus, goals, and performance. The study theorized the construct of *attractors*, which are affective experiences that earn importance and priority to a developer's cognitive system. Attractors were theorized to have the biggest impact on development performance. Finally, the study suggested that interventions (e.g., facilitating reconciliation between developers who are angrily arguing) can mediate the intensity of existing negative affect and reduce their intensity and disruption.

Wrobel [70] conducted a survey with 49 developers, assessing the participants' emotions that were perceived to be those influencing their productivity. The results showed that positive affective states were perceived to be those enhancing development productivity.

Frustration was perceived as the most prevalent negative affect, as well as the one mostly deteriorating productivity.

Ortu et al. [56], Destefanis et al. [11], and Mäntylä et al. [51] conducted a series of mining software repositories studies to understand how affect, emotions, and politeness are related to software quality issues. The studies showed that happiness in terms of frequent positive affect and positive emotions was associated with shorter issue fixing time. Issue priority was found to be associated with the arousal mental activation level, which is often associated with anxiety and burnout.

3 METHOD

We employed a mixed research method, comprising both elements of quantitative and qualitative research [7]. In particular, we opted to approach RQ1 with a quantitative investigation, while we addressed RQ2 with a mostly qualitative inquiry. As our aim was to learn from a large number of individuals belonging to a particular population, we considered a survey, implemented as an online questionnaire, to be the most appropriate instrument [17].

3.1 Sampling Strategy

We consider a software developer to be a person concerned with any aspect of the software construction process (such as research, analysis, design, programming, testing, or management activities), for any purpose including work, study, hobby, or passion. Generalizing to the population of software developers is a challenge, because we do not accurately know how many software developers exist in the world and how to reach them. We relied on the GitHub social coding community as a source that fits our purpose of generalization *well enough*, in line with several previous studies (e.g., [28]). GitHub is the largest social coding community with more than 30 million visitors each month [15], many of which are software developers working on open source and proprietary software ranging from solo work to companies and communities.

To obtain the contact information of GitHub developers, we retrieved related data through the GitHub Archive [38], which stores the collections of *public* events occurring in GitHub. In order to ensure a sample of high quality and variety, we obtained six months of archive data, from March 1 to September 30, 2014. We extracted *unique entries that provided an e-mail address*. We gathered 456 283 entries of contact data, which included email address, given name, company, and location of the developers, and the repository name related to the public activity. 41.7% of the data provided an entry for the *company* field.

3.2 Survey Design

We collected data and enhanced the survey in four rounds. During the first three rounds, we piloted the questionnaire design with a limited random sample of contact data (N=100 in each round). We discarded the pilots' contact data and questionnaire data from the final survey as many guidelines recommend (e.g., [48]).

The three pilot rounds allowed us to estimate and improve the participation and response rate of the study through a refinement of the questions and invitation e-mail. Through the pilot tests, we understood that we could expect a high percentage of delivered e-mails (between 97% and 98%) and that we could expect a low

participation rate (between 2% and 4%). The participation rate increased in each run.

The questionnaire used in the final survey is composed of (1) questions to collect demographic information, (2) one question carrying SPANE's 12 scale items, and (3) two open-ended questions on the causes of happiness and unhappiness (in terms of SPANE-P and SPANE-N components, see Section 2.1) while developing software. The questionnaire also provides an open-ended field for further comments and a field for optionally leaving an e-mail address for possible follow-ups[3]. The questionnaire is available in an archived online appendix [31].

For estimating the sample size required to make inferences regarding our population of developers, we evaluated the sample size estimations by Bartlett et al. [2], Krejcie & Morgan [46], Cochran [4], and Yamane [71], for a-priori statistical power. None of the authors have proposed sample size estimations for open-ended, qualitative entries. Therefore, we decided to opt for the most conservative settings, i.e., Yamane's simplified formula [71] with $\alpha = .01$ assuming a 2% response rate. Our calculations resulted in a desirable sample of N=664 complete responses, which we expected to reach with 33 200 requests under a 2% response rate.

We designed and published the questionnaire with *eSurvey Creator* and invited the participants via e-mail. We did not share the survey elsewhere.

3.3 Analysis and Data Cleaning

In order to answer RQ1, we needed to describe the distribution of the SPANE-B happiness score (see Section 2.1) and to provide an estimation for the population mean and median. We expected to employ non-parametric methods for the mean and median estimation, given our earlier study [33] and the information obtained from the three pilot runs.

In order to answer RQ2, we developed a coding strategy for the open-ended questions. We applied open coding, axial coding, and selective coding, as described by Corbin and Strauss [5], as follows. The first three authors individually open coded the same set of 50 random responses using a line-by-line strategy. We met through online video calls in order to compare the coding structure and strategy to reach an agreement, that is, a shared axial coding mechanism. Our unit of observation and analysis, the individual developer, was the starting point. We framed our construction of theoretical categories based on Curtis et al. [8] model of constructs that are *internal* or *external*, with the internal group being the developer's own being and the external group having the artifact, process, and people as subcategories. Then, we divided the responses and proceeded to open code them (each coder coded a third of the answers). We held a weekly meeting to follow progress and further discuss the coding structure and strategy. Finally, we merged the codes and performed a final selective coding round. We used *NVIVO 11* for the entire qualitative task. We provide a working example of the various coding phases in the online appendix [31].

Data cleaning happened during all stages of the study. We adopted common data cleaning strategies, such as outlier analysis (for example, we examined birth years after 2000 and excluded

---
[3]We designed the invitation e-mail and questionnaire text ensuring informed consent by the participants.

two 1-year old participants), and excluding participants who were not in the intended population or put random text in the text fields. We list the data inclusion and exclusion criteria in the archived online appendix [31]. We used R [58] scripts for supporting and automating the data cleaning, data exploration, and data analysis steps.

4 RESULTS

This section details the results of our investigation. We first provide descriptive statistics on the sample demographics. Then, we provide the results related to each research question.

4.1 Descriptive Statistics

Following our conservative strategy, we randomly sampled 33 200 entries from our contact list. Our sending tool delivered 31 643 (96.6%) invitation e-mails; the remaining addresses were either malformed or bounced. 2 220 individuals participated (7% response rate). 1 908 participants provided valid data for answering RQ1 (86%) while 1 318 provided valid data to provide answers to RQ2 (59%). Based on the pilots, we anticipated that some participants would leave the open-ended questions unanswered. Our sampling strategy paid off: we exceeded the required threshold (N=664) for generalizing. The rich sample offered us the opportunity to stay conservative for analyzing the data, too. We could minimize bias by retaining only the data provided by the participants that completed the entire questionnaire. That is, we kept N=1318 for answering all our RQs.

Our sample of N=1318 responses resulted in 1 234 male participants (94%) and 65 female (5%). The remaining 19 participants declared their gender as *other / prefer not to disclose*. The average year of birth was 1984 (standard deviation (sd)=9.33), while the mean was 1986. There was diversity in terms of nationality, with 88 countries. The most represented nationalities were American (24%), Indian (6%), Brazilian (6%), Russian (5%), and British (4%).

A total of 993 (75%) of the participants were professional software developers, 15% of the sample were students, and 8% were in other roles (such as manager, CEO, CTO, and academic researcher). The remaining participants were non-employed and not students.

The participants declared an average of 8.29 years (sd=7.77) of experience with working with software development, with a median of 5 years. 240 participants developed software either as a hobby, passion, or volunteer without pay, 161 participants worked either as freelancer or consultant in companies, 105 participants were a one-person company or self-employed in a startup, and 812 were employed in a company or a public organization. The reported size of the participants' company or organization also varied considerably, with 13.3% of the participants working alone, 33.6% in small entities (2-10 persons), 34.4% in medium entities (11-250), and 18.7% in large to very large entities (250-5000 and more).

Regarding the qualitative data, we reached a total of 590 codes in the initial coding phases. After the merge and cleanup phases, we obtained 219 codes that were referenced 2 280 times in text (average of 10.41 references per code).
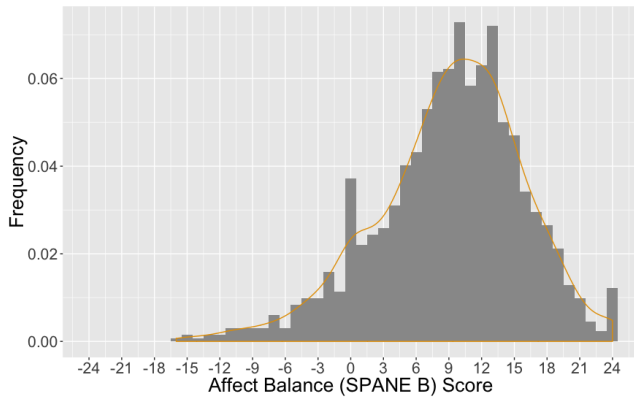
Figure 1: Happiness of software developers (SPANE-B distribution).

4.2 RQ1—What is the Distribution of (Un) Happiness Among Software Developers?

Our sample of N=1318 participants had a SPANE-B (see Section 2.1) mean score of 9.05 (sd=6.76), a median score of 10, and a range of [-16, 24].

We followed the recent suggestion by Kitchenham et al. [45] to use kernel density plot instead of boxplots. The plot of the SPANE-B score is shown in Figure 1. The plot indicates a likely non-normal distribution of the data, as expected. A description of the SPANE-B score by the *psych* R package [60] showed a skew of -0.53 and a kurtosis of 0.46, indicating a slightly asymmetrical distribution with a long tail to the left that is flatter than a standard normal distribution. Strong evidence for non-normality of the data was supported by a Shapiro-Wilk test for normality ($W = 0.98, p < 0.0001$).

We estimated the population's true mean for SPANE-B via bootstrapping as 9.05 (2000 replications, 95% CI [8.69, 9.43]). We estimated the population's true median for SPANE-B via bootstrapping as 10 (2000 replications, 95% CI [9.51, 10.71]). We show in the online appendix [31] that estimating those values with the expanded sample of N=1908 would yield similar results.

4.3 RQ2—What are the Experienced Causes for Unhappiness Among Software Developers While Developing Software?

We plotted the demographic data gathered from the questionnaire, and compared it with the SPANE-B value. None of the quantitative data plots indicated a relationship with the happiness of developers. This includes variables such as gender, age, nationality, working status, company size, percentage of working time dedicated to developing software, and monthly income. Thus, we conclude that they are not the primary determinants of unhappiness. This further confirmed our original research design to use qualitative data to explore the causes. We identified 219 causes of unhappiness, which are grouped into 18 categories and sub-categories (including the top category of causes of unhappiness while developing software). We report here only the main emerged categories and top 10 factors.

Table 1: Categories for External Causes of Unhappiness

Main category	Sub-categories
People (416)	Colleague (206) Manager (122) Customer (49)
Artifact and working with artifact (788)	Code and coding (217) Bug and bug fixing (194) Technical infrastructure (151) Requirements (99)
Process-related factors (544)	No sub-categories
Other causes (95)	No sub-categories

Because of space limitations, we provide the demographic data plots and our complete coding as open data in the online appendix [31].

4.3.1 Main Categories. The main types of factors causing unhappiness among software developers are organized under two main categories. The causes of unhappiness internal to individual developers, directly related to their personal states, or originated by their own behaviors, are classified under the *developer's own being* category. These occurred a total of 437 times. In contrast, *external causes* are the causes of unhappiness external to individual developers, by which developers are affected but have little or no control of. The total occurrence of external causes is 1 843 times. This indicates that developers are much more prone to experiencing and recalling externally-provoked unhappy feelings than internally generated ones.

The **developer's own being** (i.e., internal causes) category contains 22 internal factors. These factors do not demonstrate a clear structure. This to some extent reflects the versatile states of mind of developers and the feelings they could have while they develop software.

The factors in the **external causes** category are further divided into the sub-categories shown in Table 1. *People-related factors*: the external causes of unhappiness related or attributable to people whom developers interact with, to their characteristics or behaviors. These occurred 416 times and are further divided based on the roles of the people. *Artifact and working with artifact*: the external causes of unhappiness related to artifacts in software development projects and developers' interactions with them occurred 788 times. The causes are further grouped based on the types of artifacts that developers are dealing with. *Process-related factors*: the external causes of unhappiness related to issues in the management of software development process and day-to-day work. This type of causes occurred 544 times. *Other causes*: the external causes of unhappiness not classified under any of the above-mentioned "external factors" categories. These non-specific causes occurred 95 times.

4.3.2 10 Most Significant Causes of Unhappiness. We extracted a list of 10 factors that occurred most often in the survey responses as the causes of unhappiness. They are listed in Table 2.

Three of these top 10 causes are part of *software developer's own being*. **Being stuck in problem solving** is by far the most

Table 2: Top 10 Causes of Unhappiness, Categories, and Frequency

Cause	Category	Freq.
Being stuck in problem solving	software developer's own being	186
Time pressure	external causes → process	152
Bad code quality and coding practice	external causes → artifact and working with artifact → code and coding	107
Under-performing colleague	external causes → people → colleague	71
Feel inadequate with work	software developer's own being	63
Mundane or repetitive task	external causes → process	60
Unexplained broken code	external causes → artifact and working with artifact → code and coding	57
Bad decision making	external causes → process	42
Imposed limitation on development	external causes → artifact and working with artifact → technical infrastructure	40
Personal issues – not work related	software developer's own being	39

significant among the three factors. Software development is essentially composed of problem-solving activities, often intellectually demanding. It is common that developers may be stuck in coding, debugging and all sorts of other tasks. As one respondent commented: "*I feel negative when I get really stuck on something and cannot get around it*". Another respondent elaborated: "*I also thought of situations where I'm debugging some issue with the code and I can't figure out why it isn't working – when it seems like everything should work, but it just doesn't. This is definitely one of the biggest gumption traps I encounter*".

Another significant internal cause is a **feeling of inadequate skills or knowledge**, as shown in this response: "*Once I encountered hashmap, and I couldn't understand it while I knew it is important. I felt frustrated and afraid*". The inadequate feeling can be manifested as feeling unskilled in certain aspects of the work, feeling under-qualified with respect to the task given, or feeling a lack of familiarity with tools, languages, frameworks, or development methods that are used in the projects.

The third significant cause related to developer's own being is not related to work, but **personal issues**. Software developers are not living in vacuum while working on their software projects, and often non-work related, personal or private issues may affect them and cause their unhappy feelings during work. "*I never feel 100% productive when there's something from my private life bugging me. No, I'm not a robot, I'm human, and can't forget the rest of the world when I start IntelliJ up*". Among the non-work related issues, family related issues are most frequently mentioned: "*Family related issues has huge impact on my feeling while working, I feel down and can't achieve the goals I set for my work day*".

The seven remaining most significant factors are all *external*. Among *people-related* causes, the **under-performance of colleagues**, either team members, colleagues in other teams, or external collaborators, most often make developers experience negative feelings and affect their work consequently. An illustrative episode is reported in this response: "*Last time I felt angry when a senior developer again committed an update ruining a beautiful generic solution I've made before. It was easy to refactor it, but his ignorance or routine annoyed me*". Software development is often teamwork. It is frequently frustrating to a developer to see that other colleagues do not spend time to keep up to speed with modern development technology and practices.

It comes as no surprise that **bad code quality and coding practices** make developers unhappy. In almost all cases, bad code was a cause of unhappiness if it was written by other developers: "*Sad/angry when reading others' code that I have to use and I realize it is full of bugs*"; "*having encountered a particularly bad (unreadable, poorly formatted, not commented at all, badly structured) piece of code written by another developer that I had to work on*". Only a few participants reported unhappiness caused by "*poorly written code (often by past-me)*". That is, unhappiness from code written by the participants themselves was raised only when regretting past code.

Another significant factor related to code that makes developers feel bad is when they could not explain why the code is not working as it is supposed to (**unexplained broken code**): "*When you haven't changed the code, and suddenly the project doesn't compile anymore. Worst feeling ever. (Afraid/sad/angry)*".

Apart from code, issues in the technical infrastructure a software project relies on often contribute to negative feelings among developers, especially when it is supposed to support software development, but instead imposes constraints or limitations (**imposed limitation on development**). One respondent described this situation perfectly: "*Angry happens quite often because tools, programming languages, etc. don't do as expected. Sometimes because they are buggy, sometimes there are some limitations in them (by design/by ignoring or not considering enough use cases, etc.), which makes one need to find work-arounds/mess-up otherwise clean code, repeat code unnecessarily etc*".

Regarding the top significant causes related to general software development *process*, the respondents consider that high **time pressure** they feel, often generated by "*unrealistic*", "*unjustified*" and "*crazy*" deadlines, will almost surely push them into very unhappy states. A respondent described vividly this situation: "*I remembered a day. I have a lot of phone call from my boss to done a project. in that situation time was running and project move slowly and phone every a minute ringed*".

Contrasting the image of high time pressure, with hectic rushing towards deadlines, is working on **mundane or repetitive tasks**, which is another process-related factor that often causes negative feelings of developers. "*Tedious*", "*boring*", "*dull*", "*monotonous*", "*trivial*", "*recurrent*", etc., are the words the respondents used to describe the tasks that make them unhappy. "*I tend to feel negative*

or bad when I am doing something that is not challenging, I instantly feel sleepy and bored", a respondent stated.

Bad decision making is yet another process-related factor that often leaves developers in an unhappy state. More than bad business decisions, developers are more often affected (emotionally as well) by bad technical decisions made by their superiors or peers. One example depicts such a scenario: "*Generally negative emotions stem from board meetings where an executive or coworker makes an uninformed or ill-advised decision that causes a 'dirty' codebase change*". Bad decision making is also perceived by developers if they are not involved in decision making processes.

5 DISCUSSION

Through our analysis to answer RQ1 (Section 4.2), we estimated the population mean SPANE-B score to be 9.05, indicating a happiness balance on the positive side (see Section 2.1). In terms of the norms reported in Diener et al. [13], this result is in the 65th percentile, indicating that the happiness balance is also higher than what could be expected in a larger human population. The various psychometric studies of SPANE report sample means but no confidence intervals, meaning that the best comparison possible is through means and standard deviations. Those studies have found mean SPANE-B scores above zero, but several score points lower than in our sample: 7.51 (sd=8.21) in a sample of men and 4.53 (sd=8.17) in a sample of women in Italy [6]; 6.69 (sd=6.88) in a sample of college students from five universities and colleges in USA and one university in Singapore [13]; 6.66 (sd=8.18) in large sample of more than 21 000 employees in the Chinese power industry [49]; 5.96 (sd=6.72) in a multicultural student sample at a large South African university [16]; 4.41 (sd=7.79) in a sample of full-time employees and 5.10 (sd=7.54) in a sample of university students, both in Portugal [63]; and 4.30 (sd=7.50) in a sample of Japanese college students [64].

Our findings about the higher-than-expected SPANE-B score confirm and reinforce our previous observations [33] that (1) software developers are a slightly happy population, and that (2) there is no evidence that the distribution of SPANE-B scores for the population of software developers should cover the full range of [−24, +24]. This does not mean that software developers are happy to the point that there is no need to intervene on their unhappiness. On the contrary, we have shown that unhappiness is present, caused by various factors and some of them could easily be prevented. Our observations and other studies show that unhappiness has a negative effect both for developers personally and on development outcomes. Furthermore, these results have implications for research, as outlined below.

For answering RQ2, we have shown a wide diversity and weight of factors that cause unhappiness among developers (Section 4.3). The causes of unhappiness that are external to developers, and thus more controllable by managers and team leaders, could have an incident rate that is 4 times the one of the factors belonging to the developer's own being. We expected that the majority of the causes of unhappiness would come from human related considerations (416 references); however, technical factors from the artifact (788) and the process (544) dominate the unhappiness of developers,

highlighting the importance of strategic architecture and workforce coordination.

Being stuck in problem solving and time pressure are the two most frequent causes of unhappiness, which corroborates the importance of recent research that attempts to understand them [35, 51, 54]. Lack of experience could explain the prevalence of the first category in some cases, but since software development is inherently about problem solving, and realistic projects include an element of problem solving and learning, it does not seem adequate to explain this result by lack of experience alone. It may be necessary to accept that software development comes with its share of difficult tasks that cannot be avoided. Psychological *grit* could be an important characteristic to train among software developers. Strategies for both coping with the negative feeling associated with being stuck and systematic strategies for actually solving problems in general and specific scenarios can be called for.

Several top causes are related to the perception of inadequacy of the self and others, which encourages recent research activities on intervening on the affect of developers [35]. Finally, we see that factors related to information needs in terms of software quality and software construction are strong contributors to unhappiness among developers. This reinforces recent research activities on those aspects (e.g., [25]) and encourages proposed [29] research activities that attempt to merge affective reactions and information needs.

5.1 Limitations

We designed our survey with the stated aim of gaining understanding of the characterization of the unhappiness of developers and the causes of unhappiness in software development. We phrased the questions in our survey by following guidelines from the literature [7, 55] and from our prior experience with the research topic [18–20, 32–37]. We phrased the questions to avoid priming specific answers to the respondents. The validation of the questions was through (1) adopting a psychometrically validated measurement instrument for happiness [13], (2) limiting the remaining quantitative questions to a demographic nature, and (3) conducting three pilot runs. We discuss specific threats to validity below.

5.1.1 Internal Validity–Credibility. With respect to the happiness measurement, as reported in Section 2.1, several large scale studies have found good psychometric properties (reliability and validity) for SPANE [6, 13, 42, 49, 63], and the instrument was empirically shown to be consistent across full-time workers and students [63] and memory recall of events [13, 49].

In order to classify the causes of unhappiness of developers, we used a qualitative coding process. Whether causality can be inferred only by controlled experiments is a much-debated issue [7, 14, 27]. Several authors, e.g., [27], maintain that human-oriented research allows causality to be inferred from the experience of the participants through qualitative data analysis, provided that there is a strong methodology for data gathering and analysis. In this case, our aim was to uncover causes of unhappiness as experienced by developers themselves. Since we extracted the causes from first-hand reports, they should accurately represent the respondents' views. As far as possible, we have remained faithful to these views when categorizing the material. The chain of evidence from source

material to results is fully documented and traceable (see the online appendix [31]). The ratio between reported internal and external causes may be affected by the respondents' ability to correctly attribute their unhappiness. We note that we claim no general relationship between any specific causes and unhappiness; only experienced causes of unhappiness are claimed.

A question-order effect [62] could have influenced the responses by setting their context. As the present study was conducted in the context of a larger study on both happiness and unhappiness, we randomized the order of appearance of the questions related to affectiveness, thus limiting a potential order effect.

Social desirability bias [26] may have influenced the answers in the sense that participants would attempt to appear in a positive light before the eyes of the enquirer. We limited the bias by informing the participants that the responses would be anonymous and evaluated in a statistical form, and addressing the ethical concerns of the study. In our view, the responses appear candid, indicating that participants have felt comfortable expressing their true views.

5.1.2 Generalizability–Transferability. Our dataset of software developers using GitHub is limited with respect to representativeness of the average software developer. The data set used in this study (see Section 3.1) contains only accounts with public activity during a six-month period. However, it is likely that a significant portion of the inactive accounts are not of interest to this study, as we sought active developers.

The degree to which our conclusions are generalizable beyond the GitHub population may be limited. For instance, it is possible that the GitHub population is slightly younger than developers in general, and age may explain differences in the degree and nature of unhappiness. Also, the sample may be biased towards people that are more comfortable with displaying their personal performance in public, or face no other kinds of barriers to doing so (e.g., company policy). However, GitHub is a reliable source for obtaining research data, allowing replication of this study on the same or different populations. The GitHub community is large and diverse, with a claim of more than 30 million visitors each month [15], many developing open source and proprietary software, and ranging from solo work to companies and communities. Furthermore, as shown in Section 4.1, our sample is well balanced in terms of demographic characteristics, including participant role, age, experience, work type, company size, and student versus worker. By comparing confidence intervals, we did not observe significant differences in terms of the SPANE-B score when varying role (worker, student) or age. This further highlights the validity and reliability of the SPANE measurement instrument and the stability of our dataset.

Our sample is not evenly balanced in terms of gender, with males being in the vast majority. We believe, however, that our sample is representative to some extent in terms of gender as well, since males are overrepresented in software engineering jobs, likely due to gender bias [23, 57, 65]. However, our sample may be extreme

in this respect; while exact data is difficult to obtain, some non-academic surveys have shown, e.g., 7.6%[4], 16%[5], and 20%[6] females, but the numbers can depend on the definition of developer and the countries or cultures represented in the data. A possible explanation is that males are particularly overrepresented among GitHub developers, but more demographic data would be needed to ascertain this. In summary, we consider our sample to be large and diverse enough to warrant claims regarding software developers to the extent possible in a single study. Further replication is necessary to validate the findings and obtain details on demographic subgroups.

5.2 Recommendations for Practitioners

Our study has found a plethora of causes of unhappiness of developers that are of interest to practitioners regardless of their roles. We summarized the most prominent ones in the present paper, but practitioners could be interested in the complete list of factors and occurrences that is freely available online as open data [31].

Team members may be interested in the causes of unhappiness for enabling self-regulation and emotional capability mechanisms [1] for reducing personal and group unhappiness. Knowing what might cause unhappiness in the short and long term could encourage developers to be more considerate towards their peers. For example, it might be worth thinking twice about leaving others to clean up badly written code. For similar reasons, managers should carefully attempt to understand the unhappiness of developers using the present paper as support. Those in leadership positions should attempt to foster happiness by limiting unhappiness. Previous research (e.g., [11, 32, 33]) has shown that the benefits of fostering happiness among developers are substantial especially in terms of software development productivity and software quality. In a related paper on the consequences of unhappiness of developers, we found that addressing unhappiness could limit damage on different aspects of software development, including developers, artifacts, and development processes [30]. We believe that the results of the present study will potentially enhance the working conditions of software developers. This is corroborated by previous research [35] suggesting that intervening on the affect of developers may yield large benefits at low cost. We note that such interventions should consider issues of privacy and cultural differences. Whether to intervene in issues outside the work context is an open question, with possible legal constraints.

Furthermore, the vast majority of the causes of unhappiness are of external type. Since external causes may be easier to influence than internal causes, and since influencing them will impact several developers rather than only one at a time, this suggests that there are plenty of opportunities to improve conditions for developers in practice.

---

[4]StackOverflow Developer Survey 2017, https://stackoverflow.com/insights/survey/2017
[5]LinkedIn Blog post "Women in Software Engineering: The Sobering Stats", reporting statistics based on LinkedIn data, https://business.linkedin.com/talent-solutions/blog/2014/03/women-in-engineering-the-sobering-stats
[6]Bureau of Labor Statistics Current Population Survey: Annual averages, Software developers, applications and systems software, https://www.bls.gov/cps/cpsaat11.htm

5.3 Implications for Researchers

We believe that the results of the present work can spawn several important future research directions. A limited set of the found causes of unhappiness has been investigated previously in the software engineering literature. However, while the previous work offers valuable results, it appears limited either because of being framed too generally in psychology research – resulting in findings regarding general job performance settings – or due to a narrow focus on a single emotion (e.g., frustration). The framing offered by the present study sheds new light on these previous studies by considering them in terms of happiness and affect. Here, we suggest three implications for research that we believe are of high importance and priority.

Our result regarding the distribution of happiness among developers suggests that happiness – in terms of the SPANE instrument score – is centered around 9.05, higher than what may be expected based on other studies using the instrument. Our question for future research is to understand whether a) a higher relativity should be embraced when analyzing the affect of developers and its impact on software engineering outcomes, or b) developers require tailored measurement instruments for their happiness as if they are a special population. Validating the score through replication, and, if it is found to be stable, investigating the reasons for it being higher than in several other populations, are important aims for future research.

As reported in the previous section, most causes of unhappiness are of external type and they may be easier to influence than internal causes. We see that much research is needed in order to understand the external causes and how to limit them. Further understanding of the underlying reasons for the ratio between external and internal causes is also needed.

Finally, the present study highlights how studies of human aspects in software engineering are important for the empirical understanding of how software development can be improved. Many questions in software engineering research require approaches from behavioral and social sciences; we perceive a need in academic discourse to reflect on how software engineering research can be characterized and conducted in terms of such paradigms.

Software engineering studies on human factors often call for further human aspects studies. Yet, we believe that the present study calls for much technical research as well, because the highest source of unhappiness among software developers is related to artifacts and working with artifacts. One example is related to debugging and bug fixing, as they appear often in the causes of unhappiness. This suggests that much research is needed for supporting humans in the maintenance of software, e.g., in terms of information needs and mechanisms for strategic coordination of the workforce and the software architecture. Furthermore, emotional support for the sometimes frustrating and tedious work with software maintenance might increase the quality of results.

6 CONCLUSION

In this paper, we presented a mixed method large-scale survey (1 318 complete and valid responses) to broaden the understanding of unhappiness among software developers. Our key contributions are as follows, and are publicly archived as open access and open data [31]:

(C1) An estimate of the distribution of (un)happiness among software developers.
(C2) An analysis of the experienced causes for unhappiness among software developers while developing software.

Our results show that software developers are a slightly happy population. The consequences of that result need to be explored in future studies. Nevertheless, the results do not remove the need for limiting the unhappiness of developers, who have repeatedly asked to be given a voice through research and in the design of studies.

The results of our study have also highlighted 219 fascinating factors about the causes of unhappiness while developing software. These should be further explored in future research and used as guidelines by practitioners in management positions and developers in general for fostering happiness on the job. We also call for replications of the study.

ACKNOWLEDGMENT

REFERENCES

[1] A E Akgün, Halit Keskin, John C Byrne, Ayse Gunsel, Ali E Akgun, Halit Keskin, John C Byrne, and Ayse Gunsel. 2011. Antecedents and Results of Emotional Capability in Software Development Project Teams. *Journal of Product Innovation Management* 28, 6 (2011), 957–973.

[2] James E Bartlett, Joe W Kotrlik, and Chadwick C Higgins. 2001. Organizational Research: Determining Appropriate Sample Size in Survey Research. *Information Technology, Learning, and Performance Journal* 19, 1 (2001), 43–50.

[3] A Ben-Zeév. 2010. *Are negative emotions more important than positive emotions?* Number 7. Psychology Today.

[4] W G Cochran. 1977. Sampling Techniques. John Wiley & Sons, New York, USA.

[5] Juliet M Corbin and Anselm L Strauss. 2008. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3 ed.). Sage Publications, London.

[6] Giulia Corno, Guadalupe Molinari, and Rosa Maria Baños. 2016. Assessing positive and negative experiences: validation of a new measure of well-being in an Italian population. *Rivista di psichiatria* 51, 3 (2016), 110–115.

[7] John W Creswell. 2009. *Research design: qualitative, quantitative, and mixed method approaches* (3 ed.). Vol. 2. Sage Publications, Thousand Oaks, California.

[8] Bill Curtis, Herb Krasner, and Neil Iscoe. 1988. A Field Study of the Software Design Process for Large Systems. *Commun. ACM* 31, 11 (Nov. 1988), 1268–1287. DOI:https://doi.org/10.1145/50087.50089

[9] Munmun De Choudhury and Scott Counts. 2013. Understanding affect in the workplace via social media. In *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13.* ACM Press, New York, New York, USA, 303–315.

[10] Peter J Denning. 2012. Moods. *Commun. ACM* 55, 12 (Dec. 2012), 33.

[11] Giuseppe Destefanis, Marco Ortu, Steve Counsell, Stephen Swift, Michele Marchesi, and Roberto Tonelli. 2016. Software development: do good manners matter? *PeerJ Computer Science* 2, 4 (2016), e73.

[12] Ed Diener, Eunkook M Suh, Richard E Lucas, and Heidi L Smith. 1999. Subjective well-being: Three decades of progress. *Psychological Bulletin* 125, 2 (1999), 276–302.

[13] Ed Diener, Derrick Wirtz, William Tov, Chu Kim-Prieto, Dong-won Choi, Shigehiro Oishi, and Robert Biswas-Diener. 2010. New Well-being Measures: Short Scales to Assess Flourishing and Positive and Negative Feelings. *Social Indicators Research* 97, 2 (May 2010), 143–156.

[14] Yanyi K Djamba and W Lawrence Neuman. 2002. Social Research Methods: Qualitative and Quantitative Approaches. *Teaching Sociology* 30, 3 (July 2002), 380.

[15] Brian Doll. 2015. A closer look at Europe. (2015). https://github.com/blog/2023-a-closer-look-at-europe [Retrieved: 2017-01-26].

[16] Graham A du Plessis and Tharina Guse. 2016. Validation of the Scale of Positive and Negative Experience in a South African student sample. *South African Journal of Psychology* (2016). Prepublished June 27, 2016, DOI: 10.1177/0081246316654328.

[17] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*. Springer, 285–311.

[18] Fabian Fagerholm. 2015. *Software Developer Experience: Case Studies in Lean-Agile and Open Source Environments*. Ph.D. Dissertation. Department of Computer Science, University of Helsinki. Series of Publications A, Report A-2015-7.

[19] Fabian Fagerholm, Marko Ikonen, Petri Kettunen, Jürgen Münch, Virpi Roto, and Pekka Abrahamsson. 2014. How Do Software Developers Experience Team Performance in Lean and Agile Environments?. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*. ACM, New York, NY, USA, Article 7, 7:1–7:10 pages.

[20] Fabian Fagerholm, Marko Ikonen, Petri Kettunen, Jürgen Münch, Virpi Roto, and Pekka Abrahamsson. 2015. Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments. *Information and Software Technology* 64 (2015), 132–147.

[21] Cynthia D Fisher. 2000. Mood and emotions while working: missing pieces of job satisfaction? *Journal of Organizational Behavior* 21, 2 (March 2000), 185–202.

[22] Denae Ford and Chris Parnin. 2015. Exploring causes of frustration for software developers. In *CHASE '15: Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering*. North Carolina State University, IEEE Press.

[23] Denae Ford, Justin Smith, Philip J. Guo, and Chris Parnin. 2016. Paradise Unplugged: Identifying Barriers for Female Participation on Stack Overflow. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)*. ACM, New York, NY, USA, 846–857.

[24] B L Fredrickson. 2001. The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions. *American Psychologist* 56, 3 (2001), 218–226.

[25] T Fritz and G C Murphy. 2011. Determining relevancy: how software developers determine relevant information in feeds. In *ACM Conference on Human Factors in Computing Systems (CHI)*. 1827–1830.

[26] A Furnham. 1986. Response bias, social desirability and dissimulation. *Personality and Individual Differences* 7, 3 (1986), 385–400.

[27] J Gläser and G Laudel. 2013. Life With and Without Coding: Two Methods for Early-Stage Data Analysis in Qualitative Research Aiming at Causal Explanations. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* (2013).

[28] Georgios Gousios, Margaret-Anne Storey, and Alberto Bacchelli. 2016. Work Practices and Challenges in Pull-based Development: The Contributor's Perspective. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, New York, NY, USA, 285–296.

[29] Daniel Graziotin. 2016. Software quality information needs. *Research Ideas and Outcomes* 2 (2016), e8865.

[30] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, and Pekka Abrahamsson. 2017. Consequences of Unhappiness While Developing Software. In *Proceedings of the 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion '17)*. ACM, New York, NY, USA. In press.

[31] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, and Pekka Abrahamsson. 2017. Online appendix: the happiness of software developers. *figshare* (2017). https://doi.org/10.6084/m9.figshare.c.3355707.

[32] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2014. Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering. *Journal of Software: Evolution and Process* 27, 7 (2014), 467–487.

[33] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2014. Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ* 2 (2014), e289.

[34] D Graziotin, X Wang, and P Abrahamsson. 2014. Software Developers, Moods, Emotions, and Performance. *IEEE Software* 31, 4 (2014), 24–27.

[35] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2015. How do you feel, developer? An explanatory theory of the impact of affects on programming performance. *PeerJ Computer Science* 1, 1 (Aug. 2015), e18.

[36] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2015. The Affect of Software Developers: Common Misconceptions and Measurements. In *Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '15)*. IEEE Press, Piscataway, NJ, USA, 123–124.

[37] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. 2015. Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering. In *SSE 2015: Proceedings of the 7th International Workshop on Social Software Engineering*. ACM, 25–32.

[38] Ilya Grigorik. 2012. GitHub Archive. (2012). https://githubarchive.org [Retrieved: 2017-01-26].

[39] Daniel M Haybron. 2001. Happiness and Pleasure. *Philosophy and Phenomenological Research* 62, 3 (May 2001), 501–528.

[40] Daniel M Haybron. 2005. On Being Happy or Unhappy. *Philosophy and Phenomenological Research* 71, 2 (Sept. 2005), 287–317.

[41] Watts Humphrey. 1996. *Managing Technical People: Innovation, Teamwork, and the Software Process*. Addison-Wesley Professional.

[42] Veljko Jovanović. 2015. Beyond the PANAS: Incremental validity of the Scale of Positive and Negative Experience (SPANE) in relation to well-being. *Personality and Individual Differences* 86 (2015), 487–491.

[43] Daniel Kahneman. 1999. Objective Happiness. In *Well-Being: Foundations of Hedonic Psychology*, Daniel Kahneman, Ed Diener, and Norbert Schwarz (Eds.). Utilitas, New York, NY, USA, 3–25.

[44] Iftikhar Ahmed Khan, Willem-Paul Brinkman, and Robert M Hierons. 2010. Do moods affect programmers' debug performance? *Cognition, Technology & Work* 13, 4 (2010), 245–258.

[45] Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong. 2017. Robust Statistical Methods for Empirical Software Engineering. *Empirical Software Engineering* 22, 2 (2017), 579–630. DOI:https://doi.org/10.1007/s10664-016-9437-5

[46] Robert V Krejcie and Daryle W Morgan. 1970. Determining Sample Size for Research Activities. *Educational and Psychological Measurement* 30, 3 (Sept. 1970), 607–610.

[47] Per Lenberg, Robert Feldt, and Lars-Göran Wallgren. 2015. Behavioral software engineering. *Journal of Systems and Software* 107, C (Sept. 2015), 15–37.

[48] Andrew C Leon, Lori L Davis, and Helena C Kraemer. 2011. The role and interpretation of pilot studies in clinical research. *Journal of Psychiatric Research* 45, 5 (2011), 626–629.

[49] Feng Li, Xinwen Bai, and Yong Wang. 2013. The Scale of Positive and Negative Experience (SPANE): psychometric properties and normative data in a large Chinese sample. *PloS one* 8, 4 (4 2013), 1–9.

[50] S Lyubomirsky, L King, and E Diener. 2005. The benefits of frequent positive affect: does happiness lead to success? *Psychological bulletin* 131, 6 (2005), 803–855.

[51] Mika Mäntylä, Bram Adams, Giuseppe Destefanis, Daniel Graziotin, and Marco Ortu. 2016. Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity?. In *MSR '16: Proceedings of the 13th International Conference on Mining Software Repositories*. Brunel University, ACM, New York, New York, USA, 247–258.

[52] A M Marino and J Zabojnik. 2008. Work-related perks, agency problems, and optimal incentive contracts. *The RAND Journal of Economics* 39, 2 (June 2008), 565–585.

[53] Tim Miller, Sonja Pedell, Antonio A Lopez-Lorca, Antonette Mendoza, Leon Sterling, and Alen Kerinan. 2015. Emotion-led modelling for people-oriented requirements engineering: The case study of emergency systems. *The Journal of Systems and Software* 105 (2015), 54–71.

[54] Sebastian C Muller and Thomas Fritz. 2015. Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. IEEE, 688–699.

[55] A N Oppenheim. 1992. *Questionnaire Design and Attitude Measurement* (2 ed.). Bloomsbury Academic, London, UK.

[56] Marco Ortu, Bram Adams, Giuseppe Destefanis, Parastou Tourani, Michele Marchesi, and Roberto Tonelli. 2015. Are Bullies More Productive? Empirical Study of Affectiveness vs. Issue Fixing Time. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories (MSR)*. IEEE, 303–313.

[57] Marco Ortu, Giuseppe Destefanis, Steve Counsell, Stephen Swift, Michele Marchesi, and Roberto Tonelli. 2016. How diverse is your team? Investigating gender and nationality diversity in GitHub teams. *Peerj Preprints* 4:e2285v1 (2016).

[58] R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org

[59] Rakesh Rana, Miroslaw Staron, Christian Berger, Agneta Nilsson, Riccardo Scandariato, Alexandra Weilenmann, and Martin Rydmark. 2015. On the role of cross-disciplinary research and SSE in addressing the challenges of the digitalization of society. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 1106–1109.

[60] William Revelle. 2016. *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois. http://CRAN.R-project.org/package=psych R package version 1.6.6.

[61] James A Russell. 2003. Core affect and the psychological construction of emotion. *Psychological review* 110, 1 (2003), 145–172.

[62] Lee Sigelman. 1981. Question-Order Effects on Presidential Popularity. *Public Opinion Quarterly* 45, 2 (June 1981), 199–207.

[63] Ana Junça Silva and António Caetano. 2013. Validation of the Flourishing Scale and Scale of Positive and Negative Experience in Portugal. *Social Indicators Research* 110, 2 (2013), 469–478.

[64] Katsunori Sumi. 2014. Reliability and Validity of Japanese Versions of the Flourishing Scale and the Scale of Positive and Negative Experience. *Social Indicators Research* 118, 2 (2014), 601–615.

[65] Josh Terrell, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson Murphy-Hill, Chris Parnin, and Jon Stallings. 2016. Gender differences and bias in open source: Pull request acceptance of women versus men. *Peerj Preprints* 4:e1733v2 (2016).

[66] E R Thompson. 2007. Development and Validation of an Internationally Reliable Short-Form of the Positive and Negative Affect Schedule (PANAS). *Journal of*

Cross-Cultural Psychology 38, 2 (March 2007), 227–242.

[67] Jeanne L Tsai, Brian Knutson, and Helene H Fung. 2006. Cultural variation in affect valuation. *Journal of Personality and Social Psychology* 90, 2 (Feb. 2006), 288–307.

[68] Robert P Vecchio. 2000. Negative Emotion in the Workplace: Employee Jealousy and Envy. *International Journal of Stress Management* 7, 3 (2000), 161–179.

[69] David Watson, Lee A Clark, and Auke Tellegen. 1988. Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology* 54, 6 (June 1988), 1063–1070.

[70] Michal R Wrobel. 2013. Emotions in the software development process. *2013 6th International Conference on Human System Interactions (HSI)* (2013), 518–523.

[71] Taro Yamane. 1967. Statistics: An introductory analysis. Harper and Row, New York City, US.

[72] J M Zelenski, S A Murphy, and D A Jenkins. 2008. The Happy-Productive Worker Thesis Revisited. *Journal of Happiness Studies* 9, 4 (2008), 521–537.