Characterizing and Predicting Mental Fatigue during Programming Tasks

Saurabh Sarkar
Microsoft
Redmond, WA, USA
Email: saurabsa@microsoft.com

Chris Parnin
North Carolina State University
Department of Computer Science Raleigh, NC, USA
Email: cjparnin@ncsu.edu

***Abstract*—Mental fatigue reduces one's cognitive and physical abilities. In tasks requiring continuous attention, such as driving, fatigue is a well-known risk. However, when fatigued during daily tasks, such as programming, the nature of risk is more diffuse and accumulative, yet the consequences can be just as severe (e.g. defects in autopilot software). Identifying risks of fatigue in the context of programming can lead to interventions that prevent introduction of defects and introduce coping mechanisms. To character and predict these risks, we conducted two studies: a survey study in which we asked 311 software developers to rate the severity and frequency of their fatigue and to recall a recent experience of being fatigued while programming; and an observational study with 9 professional software developers to investigate the feasibility of predicting fatigue from interaction history. From the survey, we found that a majority of developers report severe (66%) and frequent (59%) issues with fatigue. Further, we categorized their experiences into seven effects on programming tasks, which include reduced motivation and reduced ability to handle tasks involving large mental workloads. From our observational study, our results found how several measures, such as focus duration, key press time, error rates, and increases in software quality warnings, may be applied for detecting fatigue levels. Together, these results aims to support developers and the industry for improving software quality and work conditions for software developers.**

I. Introduction

Software developers are under pressure to deliver working software [1], often at the cost of their own health [2]. A culture has permeated the software industry that emphasizes performance over well-being. The expectation of long work-hours and participation in death marches [3], places developers in a situation where they must work in a state of *fatigue* (a physiological state of reduced mental or physical capability [4]). Researchers have hypothesizes that fatigue can result in errors in software, often as a direct result of absentmindedness or exhaustion. Smith theorizes when given a strict deadline, developers will become overworked and stressed to meet that deadline and resort to shortcuts [5] that can lead to costly mistakes and technical debt [6].

Previous research has characterized the fatigue in the context of physical tasks (driving [7], nursing [8] or performing other physical activities [9]). Fatigue caused by long and demanding tasks can result in mistakes [10]. These mistakes are due to reduced cognitive functioning like distraction, low decision making power, diminished reasoning capabilities, and poor attention [11], [12], [13]. In addition to errors, workers engaged in vigilance tasks or repetitive tasks often experience "operator fatigue" or "skills fatigue", which can be observed by increased response time when performing tasks, including a slowdown in keyboard and mouse inputs [14], [15]. This lower performance can be characterized by comparing work against standards of precision, completeness, cost, and pace when at full mental capacity.

Although developers have openly complained [16], [17], [18] about fatigue in the workplace, there has been limited research to document its effects or ways to combat it. Unlike tasks like driving where constant vigilance is required, programming does not require nonstop dedication to work, instead offering opportunities for recuperation and a dynamic pace of work. Our work aims to cover this gap in understanding. When programming while fatigued, the consequences of fatigue may not be as acute or immediate as when piloting an airplane or driving an car, but the long-term effects can be just as severe (e.g. defects in autopilot software) and costly (e.g. lower productivity).

> *The goal of this research is to identify risks associated with fatigue when programming and to provide mechanisms to detect the occurrence of fatigue.*

In this paper, we present our understanding about fatigue from two studies: a survey and an observational study. We conducted a survey with 311 software developers whom we asked about their experiences with fatigue. We categorized the experiences into six effects of fatigue for software developers. Next, we conducted an observational study on the work and sleep patterns of nine professional software developers for a period of at least seven days. As the developers worked on their own tasks, we monitored and collected logs of the activities, gathering a total of 56 log entries. We asked our participants to assess their mental state and their performance per the task using the NASA-TLX task-load assessment technique [19]. The results of our study establishes the feasibility of predicting the occurrence of fatigue using features such as keyboard dynamics, navigation transition patterns, code quality, and error rates. This paper reports the primary findings from work done for a Masters thesis, which can be consulted for more details[1].

[1] https://repository.lib.ncsu.edu/bitstream/handle/1840.16/10358/etd.pdf

II. Methodology

We conducted a survey and a formative study to answer the following research questions.

- RQ1: *How severely and frequently does fatigue impact developer productivity?*
- RQ2: *What are the effects of fatigue on programming tasks?*
- RQ3: *To what extent can we predict occurrence of these effects from developer interaction history?*

We used the survey to answer to first assess the severity and extent of fatigue in the workplace (RQ1), and then to characterize different factors associated with developer productivity (RQ2). Based on these results, we designed a plugin to collect interaction history from developers and predict the occurrence of fatigue based on the derived factors (RQ3).

A. Survey

The survey[2] consisted of 11 questions grouped into categories: sleep, fatigue, work. The questions about sleep were designed to get information regarding their sleep habits and circadian rhythm. We asked several questions to learn about their views on the fatigue when programming. We derived questions related to work to get information about their work habits and performance in order to provide context for our interpretation. We posted links to this survey on Reddit groups, Quora, in Computer Science Facebook groups, and emailed out software development mailing lists. Participants' age ranged from 18–74.

To answer RQ1, we asked developers to rate the severity and frequency of the effects of fatigue according to their personal experience using a Likert scale from 1 to 5. Further, we also asked developers to identify the primary causes associated with their fatigue. The survey included a question with some possible factors identified previously [9], [10] that might cause mental fatigue: Stress, Mood, Sleepiness, Physical work, and Non-willingness towards work.

To answer RQ2, we focused on a single survey question: *Do you feel when you are tired that it influences your work? If yes, what are some examples?*

Card sorting is a technique used to create mental models and define taxonomies from data [20]. Both authors randomly selected 30 responses to this question in order to identify initial reasons in examples. We used an online[3] open card sort to group the responses to this question into categories. Problem label clarifies the theme behind their reasons. We repeated the exercise twice, once individually and the next time together. We compared the labels and finalized seven labels. We achieved Cohen's Kappa above 0.7.

B. Observational Study

To investigate the feasibility of identifying fatigue during programming tasks and recognizing its impact on programming tasks, we conducted a formative study on developers. Along with monitoring the developers' work activities, the study included a sleep test for the participant and surveys about their work. Specifically, we wanted to identify and validate the observed factors identified by the survey from their interaction history.

Nine professional software developers participated in the formative study. Participants were recruited from online freelancing websites, on the basis of their daily work routine and expertise with the Eclipse IDE. On an average, the participants have an industrial experience of 3.5 years. They all work on software development around 8–9 hours a day. The study was conducted for a period of about 7 days. The participants were instructed to perform their normal development work; however, they would be occasionally required to fill out a survey.

C. Prediction Measures

We implemented a plug-in, DevFatigue[4], for the Eclipse IDE, in order to record interaction data and administer the survey after each programming session as well as an end-of-day survey. We recorded keystroke and mouse dynamics, focus events, and project events.

Performance Measures: Programmers are less likely to be involved in very repetitive tasks for many hours without break, still, we may be able to observe similar decreases in performance as seen with operators.

- *Keydown Time:* Time spent between holding down a key.
- *Time between Keys:* Time spent between two consecutive key up and key down events.
- *Mouse Velocity/Acceleration:* Velocity and acceleration of the cursor.
- *Time between Clicks:* Time spent between two consecutive mouse up and mouse down events.

Error Measures: Since previous literature on general computing tasks suggests an increase in the number of errors per key [14], [15] when fatigued, we expect to observe a similar occurrence for software development activities.

- *Corrections per Key Pressed:* Number of times a correction (backspace/del/undo) is made per key pressed.

Quality Measures: We used Checkstyle, a static analysis tool which detects coding standard violations, to evaluate the code quality. Checkstyle generates results in terms of errors in every file.

- *Warnings Delta:* We calculated code quality considering the change in the code analysis errors in a project for each day over the baseline recorded from the previous day.

Focus/Mental Workload Measures/Motivation: We expect that when a developer is more fatigued, he/she will have a decreased ability to maintain focus on activities. One way to measure this is by evaluating time spent per files. Further, we

[2] https://github.com/alt-code/Research/blob/master/Fatigue/Survey.md
[3] https://conceptcodify.com/
[4] http://www4.ncsu.edu/~ssarkar4/fatigue/eclipse/updatesite/

expect developer to work on less number of files in a fatigued state (have a reduced workload). Finally, we expect fatigued developers to have have an increase in "easier tasks" such as refactoring work, and a reduced overall activity level.

- *Focus Duration:* Time spent per file.
- *File/Method Transition:* Amount of files and methods interacted with during programming session.
- *Systematic manipulation:* For this analysis, we defined systematic manipulation as the functions provided by the tools in an IDE, such as copy/paste, refactoring, etc. for ease in structural editing.

D. Analysis

Developers' work environments were instrumented and experience sampling was used for labeling sessions and days with self-assessed fatigue levels. For analyzing the relation of the features with fatigue, we used the fatigue score reported by the developers in the DevFatigue daily survey responses. The fatigue was scored using a scale from 0 to 10 and we divided the scores into three level: score 0–3 represents Low, 4–6 represents Medium and 7-10 represents High fatigue levels. For each factor, along with empirical analysis we found out the Spearman's rank correlation coefficient of the group mean with respect to the fatigue level. The correlation coefficient supported our analysis to measure the strength of association between the features and fatigue.

III. Results

We received survey responses from 311 software developers and interaction history collected over 56 days for 9 participants.

A. RQ1: How severely and frequently does fatigue impact programmer productivity?

66% of developers rated the severity of fatigue during programming tasks as high or very high. 59% of developers rated the frequency of their fatigue during programming tasks as often or very often. *Stress* and *sleepiness* were the most voted causes. Developers were fatigued due to poor sleep habits and the stress of working long and demanding hours. Together these factors are consistent with our definition of mental fatigue in programming tasks.

In summary, we found that a majority of the developers considered fatigue to be a severe and frequent.

B. RQ2: What are the effects of fatigue on programming tasks?

We asked participants to characterize the impact of fatigue on their programming tasks in an open-ended response. After a card sorting analysis, we identified seven effects. Factors are ordered from the highest to lowest frequency of cards. A response from the survey that best represents the category is listed in each description.

- **Performance:** [28.42%] *"When I am tired I cannot come up with solutions as quickly, or sometimes not at all..."* Fatigued developers perform work slower. Suboptimal approaches and having less reasoning capabilities contribute to degradation of performance.
- **Focus:** [20.64%] *"When I am tired I find myself drifting from tasks that require a lot of focus (mental effort)".* Fatigued developers have reduced attention and increased absentmindedness. Respondents were concerned about being easily distracted and less productive. Impaired concentration leads to lack of regard for details, resulting in lower quality, and ability to stay on task.
- **Productivity:** [19.03%] *"It harms my productivity. It also stifles creativity so you might not come up with the best solution the first time. The quality of my works declines drastically..."'* Fatigued developers are less likely to produce code that satisfies a developers' intended goals and standards. Apart from poor reasoning, fatigue also affects quality and creativity. Responses also acknowledge an intersection between performance and productivity.
- **Errors:** [10.72%] *"Introducing bugs into code, not deploying latest binaries and being surprised at lack of change in behavior. Tendency to cut corners, put things off and make mistakes..."* Fatigued developers make mistakes and reduce quality of code. Distraction and lack of focus is also contribute to increased chances for making logic errors and silly mistakes.
- **Mental Work-set:** [7.77%] *"I cannot keep large coding context in my memory. I zone out all the time..."* When fatigued, a developer's mind looses the ability to focus on more complex tasks and ability to coordinate concurrent activities. Developers avoid complex tasks and focus on unnecessary trivial tasks.
- **Motivation:** [7.77%] *"When I'm tired I lose all desire to continue with the task at hand which is upsetting. More resistant to starting new work, particularly if it seems 'hard'..."* Fatigue induces unwillingness to work. Responses acknowledged that without motivation to work, developers move towards delaying the work, in fact, they are often unwilling to even start.

C. RQ3: To what extent can we predict occurrence of these effects from developer interaction history?

The data collected during the observational study is summarized in Table I.

TABLE I
DATA COLLECTED OVER 56 DAYS FOR 9 PARTICIPANTS.

User	Low	Med.	High	TLX	Activities	Time (Mins.)
1	3	5	0	33.1	9855	863.18
2	5	2	1	25.1	13822	346.95
3	2	3	3	47.2	2110	134.23
4	8	0	0	24.7	21584	662.76
5	2	1	1	30.0	3982	113.33
6	3	2	1	37.2	3079	233.75
7	3	0	0	41.5	1546	54.76
8	3	2	1	36.0	6566	743.11
9	2	2	1	24.5	5708	155.16
Total	31	17	8	33.27	68252	3307.23

Performance Measures: We expected to observe decreases in performance measures associated with high fatigue days.

- *Slower Key Movements:* The most common pattern we observed that on a low fatigue day fast responses (<1500 milliseconds) covered around 97% of all the key-down times. In case of medium fatigue level, the fast responses cover only 90% of the distribution and in case of high fatigue level, fast responses covers around 80% of the distribution. We also calculated the Spearman's rank correlation coefficient comparing the key-down time with respect to the fatigue score associated with them. We found out the coefficient as 0.99. Similiar patterns can be observed for Time between Keys.
- *Less Reliable Mouse Movements:* There were more mouse events (clicks, scrolls, etc.) in a higher fatigued level, while in lower fatigue days, mouse usage was reduced. The time between clicks had similiar predictive power as time between keys (.99). However, Mouse Velocity, Acceleration were not as strongly correlated (-.78 and -.62). Some possible explanations include: different mouse devices and usage may vary, and mouse events occur more intermittently than key movements in low fatigue state.

Error Measures: We expected to observe increases in error measures associated with high fatigue days.

- *More errors, but not as many as we expected:* While we did observe an increase in error rates (15%) on low fatigue vs high fatigue days (30%), this was not necessarily a linear relationship. For example, the average error rate on medium fatigue days was (34%). We also calculated the Spearman's rank correlation coefficient comparing error rate for keys with respect to the fatigue score associated with them. We calculated the Spearman's Correlation Coefficient to be 0.73.

Quality Measures: We expected to observe decreases in quality measures associated with high fatigue days.

- *Some decrease in quality observed:* Our hypothesis is supported by the Spearman's correlation coefficient, that is, -0.92 which represents a significant decrease in code quality with increase in fatigue level. We found that the amount of Checkstyle warnings increased when a developer was in a higher fatigue state, suggesting that there may be instances of less quality control.

Focus/Mental Workload Measures:

We expected to observe decreases in focus duration as well as decreases in amount of artifacts worked with when fatigued.

- *Less focus duration, but more artifacts when fatigued:* Developers on average focused 9 minutes on a file (low fatigue) vs. 3 minutes on a file (high fatigue). The calculated Spearman's correlation coefficient is -0.91 which supports our hypothesis that in case of fatigue a developer would have a decreased focus duration.

  Surprisingly, we observed an increase in artifacts worked with when fatigued. However, this may have been a result in an increase in browsing behavior and a decrease in editing behavior or an increase in navigation errors which resulted in viewing the wrong files.

Motivation:

We expected to observe increases in "easy work" and decreases in activity when fatigued.

- *More systematic tasks when fatigued:* We observed an increase in the amount of systematic tasks (refactoring, copy/paste) when fatigued (.99). There is an average 97 commands associated with systematic tasks (low fatigue) vs. 215 commands (med fatigue), and 235 commands (high fatigue). We believe this is consistent with lower desire to work on "harder tasks" when fatigued as reported in our survey.
- *Less overall activity when fatigued:* When fatigued, we observe an overall decrease in the amount of user activities, key usage, mouse usage and time spent on software development. This is consistent with lower motivation reported in our survey, which makes developers work less.

IV. Discussion

A. Limitations

From our survey and study, we can report some initial evidence on the effects of fatigue during programming tasks and identified several promising features for detecting fatigue from interaction history. However, fatigue is a complex phenomenon. While we observed often large differences in means between self-reported fatigue levels, in our formative study, we lack sufficient data to conclude statistical significance in those differences. Further, accounting for variance between developer behavior will require further research in how to calibrate and understand the reliability of these measures across individuals.

Understanding why certain metrics may be associated with fatigue still requires further investigation. For example, we observed slower time to perform a key press in fatigued developers, but why? While most developers rarely take more than a second to press a key, fatigued developers sometimes took 3–5 seconds to press a key. In these cases, developers may type more pausefully and experiencing a decline in ability to recall tasks and procedural operations. If a developer presses the shift key, but fails to recall the next key sequence needed for an operation in the IDE, then they may take several seconds to perform a key press.

Our approach relies on self-reported data from experience sampling methods rather on physiological sensors. The advantage is that this approach is non-invasive and non-intrusive. However, the human body has natural fluctuations in arousal through out the day [21]. For carrying out any activity, the context and environment are also very important in determining the level of fatigue during the day. For example, whether it is work-related variables such as motivation, stress, boredom, distractions; or environment-related factors such as ambiance that includes temperature, noise, lighting; or even lifestyle factors such as food, drugs like caffeine, posture, it is important to consider these contextual variables as they can contribute to the fatigue a human being experiences [22]. Finally, extreme fatigue is rare, which creates a imbalance in the dataset and may even hamper their ability to reliably report their fatigue.

B. Implications

1) Methods for coping with fatigue: In the survey, we asked the participants about what are they most likely to do when are tired at work. We gave them three options: to continue working, switch task or stop working and rest. The majority of respondents voted for "Switch to other tasks and until you regain your concentration." Furthermore, corpus of data, from the survey responses, provides inspiration for coping mechanisms and a new perspective to address the effects of mental fatigue. It could be achieved by various methods like alerts, screen freezes or the Pomodoro technique suggested by one of the respondents,

"I think slower, and start becoming distracted. But I found that using a Pomodoro Timer can keep my energy up and myself focused."

Overall, developers may need more flexibility in choosing from a variety of tasks that require different types of concentration and mental effort.

2) Detecting skill fatigue and energy levels: Skill fatigue can occur from repetitively performing demanding tasks. Developers engaging in long coding sessions may experience diminishing capacity to work as the day advanced. In these cases, continuous measures of fatigue may be able to detect when a developer's productivity is waning. Using this information, developers can more proactively learn how to manage their energy levels and rhythms during the day. Good management of the working schedule/time and effort invested in each task, along with well-placed breaks, can improve the performance with better mental health, thereby delaying the effects of skill fatigue.

Altogether, long-term measures would be able to suggestion the ideal length of a coding "sprint" for a particular developer based on their stamina as well as best coding times.

3) Software quality monitoring: In our survey, developers reported cutting corners on quality when fatigued. We also observed an increase in the amount of Checkstyle warnings generated when fatigued developers worked. This suggests that code written by fatigued developers may need to be flagged for code review and may require more quality control. Several companies have introduced employee incentive programs such as wearing a Fitbit in order to encourage more exercise. Likewise, companies may consider introducing bonus incentive programs for considering the amount of sleep developers are getting. In limited circumstance, it may even make sense to have restrict access to mission critical software if they have had insufficient sleep.

Companies may consider multiple mechanisms to prevent a decline in software quality caused by fatigued programmers.

4) Building a fatigue detector: We have explored several possible features that can be used to detect different effects of fatigue on software programming tasks. For example, different measures such as focus duration, key press time, error rates are promising features to explore. By combining, calibrating, and extending these measures, future researchers can design a prediction model for detecting fatigue according to user's work pattern. Especially promising is the idea using a combination of other physiological sensors, such as pulse detectors, skin conductance, and eye tracking devices. In future work, a modeling approach can be used to pinpoint the most influential variables in predicting fatigue, by using a linear regression or mixed model approach.

However, the main challenge moving forward will be finding a reliable and sufficient dataset of programming activity and measures of fatigue. In our dataset, a developers typically reported 1–2 days of high fatigue for every week of activity. This suggests that at least a month of developer history would need to be recorded in order to get a sufficient number high fatigue days.

V. Conclusion

To comprehend the mental fatigue of software developers, we conducted a survey and card sort to catalog effects of fatigue on programming tasks. We identified seven effects of mental fatigue, including lower performance, reduced software quality, and a reduced capacity to work on complex tasks. The survey study contained many other questions such as asking developers about their coping strategy for fatigue. Future work would include analyzing those responses and coming up with more coping mechanism. With the increased interest in the behavior of software developers, more research should be carried out to *identify* the adverse effects of mental fatigue on software development.

The formative study we conducted involved observing the work and sleep patterns of nine software developers for a period of at least seven days. Using interaction history data from the developers, we reported on the ability to detect fatigue in professional programmers with respect to all the defined effects. We found several promising features, including mean focus duration, key press time, error rates, and increases in Checkstyle warnings. We believe these measures can be considered as guidelines for building a predictor for detecting fatigue automatically while performing software development tasks.

Besides improving understanding about human factors that can affect programmers' performance, this study also makes a practical contribution as it suggests methods for programmers to improve their performance. Further, we discuss about the future studies and broader impacts to the software industry.

References

[1] H. Edholm, , M. Lidström, J.-P. Steghöfer, and H. Burden, "Crunch time: The reasons and effects of unpaid overtime in the games industry," in *Proceedings of the 39th International Conference on Software Engineering: SEIP Track*, ser. ICSE 2017, 2017.

[2] S. Klitzman, J. S. House, B. A. Israel, and R. P. Mero, "Work stress, nonwork stress, and health." *Journal of behavioral medicine*, vol. 13, no. 3, pp. 221–243, Jun. 1990. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/2213867

[3] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "What went wrong: A survey of problems in game development," *Comput. Entertain.*, vol. 7, no. 1, pp. 13:1–13:22, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1486508.1486521

[4] P. M. Muchinsky, *Psychology Applied to Work*, 6th ed. Belmont, CA: Wadsworth/Thomson Learning, 2010.

[5] H. J. Smith and M. Keil, "The reluctance to report bad news on troubled software projects: a theoretical model," *Information Systems Journal*, vol. 13, no. 1, pp. 69–95, 2003. [Online]. Available: http://dx.doi.org/10.1046/j.1365-2575.2003.00139.x

[6] R. Torkar, P. Minoves, and J. Garrigós, "Adopting free/libre/open source software practices, techniques and methods for industrial use," *Journal of the Association for Information Systems*, vol. 12, no. 1, p. 88, 2011.

[7] S. D. Baulk, S. N. Biggs, K. J. Reid, C. J. van den Heuvel, and D. Dawson, "Chasing the silver bullet: measuring driver fatigue using simple and complex tasks," *Accident Analysis & Prevention*, vol. 40, no. 1, pp. 396–402, 2008.

[8] L. M. Barker, "Measuring and modeling the effects of fatigue on performance: Specific application to the nursing profession," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2009.

[9] K. Saito, "Measurement of fatigue in industries," *Industrial Health*, vol. 37, pp. 134–142, 1999.

[10] L. Aaronson, C. Teel, V. Cassmeyer, G. Neuberger, L. Pallikkathayil, J. Pierce, A. Press, P. Williams, and A. Wingate, "Defining and measuring fatigue." *Journal Of Nursing Scholarship*, vol. 31, no. 1, pp. 45–50, 1999.

[11] P. C. Winwood, A. H. Winefield, D. Dawson, and K. Lushington, "Development and validation of a scale to measure work-related fatigue and recovery: the occupational fatigue exhaustion/recovery scale (ofer)," *Journal of Occupational and Environmental Medicine*, vol. 47, no. 6, pp. 594–606, 2005.

[12] T. Morris and J. C. Miller, "Electrooculographic and performance indices of fatigue during simulated flight," *Biological psychology*, vol. 42, no. 3, pp. 343–360, 1996.

[13] D. Kahneman, "Remarks on attention control," *Acta Psychologica*, vol. 33, pp. 118–131, 1970.

[14] A. Pimenta, D. Carneiro, P. Novais, and J. Neves, *Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 222–231. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40846-5_23

[15] ——, *Analysis of Human Performance as a Measure of Mental Fatigue*. Cham: Springer International Publishing, 2014, pp. 389–401. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-07617-1_35

[16] H. Makabee, "Effective software design," August 2011. [Online]. Available: http://effectivesoftwaredesign.com/2011/08/23/how-decision-fatigue-affects-the-efficacy-of-programmers/

[17] Dan, "Top 10 symptoms of developer burnout," 06 2011. [Online]. Available: http://tech.onthis.net/2011/06/16/top-10-symptoms-of-developer-burnout/

[18] D. Nanghaka, "Developer fatigue," August 2012. [Online]. Available: http://dndannang.blogspot.com/2012/07/developer-fatigue.html

[19] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Human Mental Workload*, ser. Advances in Psychology, P. A. Hancock and N. Meshkati, Eds. North-Holland, 1988, vol. 52, pp. 139 – 183. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166411508623869

[20] A. Begel and T. Zimmermann., "Analyze this! 145 questions for data scientists in software engineering." in *In Proceedings of the 36th International Conference on Software Engineering (ICSE 2014).*, June 2014.

[21] D. LLOYD and M. STUPFEL, "The occurrence and functions of ultradian rhythms," *Biological Reviews*, vol. 66, no. 3, pp. 275–299, 1991. [Online]. Available: http://dx.doi.org/10.1111/j.1469-185X.1991.tb01143.x

[22] S. Folkard and P. Tucker, "Shift work, safety and productivity," *Occupational medicine*, vol. 53, no. 2, pp. 95–101, 2003.