ABSTRACT

SARKAR, SAURABH. Investigation of the Effects of Mental Fatigue on Programming Tasks. (Under the direction of Christopher Parnin.)

Mental fatigue reduces one's cognitive and physical abilities. Fatigue has been considered a critical factor in tasks such as driving, but its impact on software development tasks has not been well explored. This thesis argues that this factor also applies to programming tasks. Defining a framework that explains fatigue in the context of programming can help us predict adverse effects and introduce coping mechanisms. To begin constructing such a framework, 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 emergent results in more detail. From the survey, we found that a majority of the developers considered fatigue to be a severe (66%) and frequent (59%) problem. Further, we categorized their experiences into six factors, which include reduced motivation and reduced ability to handle tasks involving large mental workloads. The observational data we collected evaluates ten research questions. For example, our results suggest how fatigue affects the user interaction during program comprehension and how a developer's transition pattern can be applied to detect his fatigue level. In this thesis, we report many predictors that can be utilized to detect fatigue level of a developer in industry. We analyze our findings and in support of the research questions provide design guidelines for detecting fatigue and building coping mechanisms. Our research aims to support developers and the industry for improving software quality, sharing best practices and paves the way for future research.

Investigation of the Effects of Mental Fatigue on Programming Tasks

by
Saurabh Sarkar

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2015

APPROVED BY:

Timothy J. Menzies

Emerson Murphy-Hill

Christopher Parnin
Chair of Advisory Committee

DEDICATION

To my parents and my family for making endless sacrifices to provide me a great livelihood and showing me the righteous path to follow.

BIOGRAPHY

Saurabh Sarkar was born and brought up in Ghaziabad, a town in the state of Uttar Pradesh, India. He received his B.Tech in Computer Science and Engineering from G. L. Bajaj Institute of Technology and Management, Uttar Pradesh, India in 2010. After working for couple of years in software industry including Tata Consultancy Services, he joined the Master's Thesis program in Computer Science in North Carolina State University. Saurabh has also done internship at Aspera (an IBM company), Emeryville, CA in Summer 2014. Saurabh will be working with Microsoft at Redmond from 2015.

ACKNOWLEDGEMENTS

I am deeply indebted to my advisor Professor Christopher Parnin for his endless guidance, motivation, and support during this thesis work. Through this work he introduced me to the world of research especially in the area of Software Engineering. I am thankful to Professor Emerson Murphy-Hill for his guidance that inspired me to enroll in the MS Thesis program. I am also thankful to my committee members for discussing and providing useful feedback for this work.

I am thankful to my colleagues (in no particular order) Brittany Johnson, Denae Ford, Anthony Elliott, Kevin Lubick, Titus Barik and Justin Smith, Manish Singh and others from AltCode/DLF/RealSearch group who have supported me in my research work through reviews, discussions and meetings. I am grateful to Anup Kalia, a PhD student at NCSU and a dear friend, for his guidance and support through out this research work and my stay in Raleigh.

I am thankful to my roommates Saransh, Anudeep and Naval and my friends (in no particular order) Vandana, Arpit, Amit, Rohit, Shashank, Snehal, Saili, Wageesha, Garima and Nirav for making my life wonderful here in Raleigh.

I am thankful to my colleagues and friends from TCS (in no particular order) Stephen, Srinivasan, Kejul, Rujuta, Paulomi, Punit, Vikram and Arun who have inspired me to do the thing I wanted and pursue my dream. Without their support I did not even think of doing a Masters in the US.

Last but not least, I am thankful to my wonderful brothers Sanjay and Rakesh and my family for providing me all kind of support to come here to the US and conquer my dreams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

There is an increasing demand for quality software to meet the varied challenges of our world. To accommodate this imperative need, software developers are under pressure to deliver. A culture has permeated the software industry that emphasizes performance over the health of the developers. The expectation of long work-hours and participation in death marches [Pet09], places developers in a situation where they must work in a fatigued state.

Errors in software can be due to low performance of developers, often as a direct result of absentmindedness or exhaustion [SK03]. For example, developers report that when given a strict deadline, they become overworked and stressed to meet that deadline, which is consistent with our survey findings. As a result of their fatigue, they struggle to concentrate and even the simplest of tasks take longer to complete. Furthermore, software developers who are under immense deadline pressure may resort to shortcuts that can lead to costly mistakes and technical debt. Unfortunately, this can also affect their peers and the company they work for when the applications they are working on have to be integrated and scaled. In addition, there is a reluctance of transmitting negative project status information to the upper ranks of the organizational members. This makes it difficult for the decision makers to take appropriate actions at the right time [SK03].

Our overall research goal is to investigate symptoms related to programming when fatigued and to identify potential ways to predict the occurrence of fatigue. Broader impacts include the potential for establishing workplace guidelines for software development work and tools for flagging code which may need further inspection.

Empirical studies have been conducted on the cognitive aspects of software engineering. Khan et. al. [Kha11] have found a relationship between a programmers' moods and the performance of programming debugging. They conducted a study asking programmers to watch a short movie clips selected for their ability to provoke specific moods and then to complete a debugging test. The study showed that there was significant difference in the programmers' debugging performance after watching low- and high-arousal evoking video clips. This suggested that programmers' moods influence some programming tasks such as debugging. For general computing tasks, researchers [Pim13] [Pim14] have proposed models to detect and monitor fatigue, based on the performance of user interaction with the computer. The focus is only on the performance indicators and not on the other effects of fatigue such as errors and code quality. Further the research focuses on mostly mundate tasks, which exhibit different patterns than we have observed with developers. Our approach is an non-invasive and non-intrusive way to monitor, collect and analyze developer's interaction and find out key indicators that can help us detect multiple symptoms of fatigue in software development related tasks.

In this thesis, we focus on understanding the rudimentary cause of quality degradation in software due to developer's fatigue by investigating developer's interaction history. Unlike other studies, we don't focus on analyzing only the key/mouse usage data available through user interaction, but also other key features like transition patterns, energy curve, code quality, developer's focus and mental work-set, etc. To gather measures of the fatigue, we support our analysis by developer's retrospection of the work using a task-load assessment technique [HS88]. We gather data and present our understanding about fatigue from two studies: a survey (Chapter 3) and an observational study (Chapter 4). The survey we conducted had 311 software developers whom we asked about their experiences with fatigue. We categorized the experiences into six effects of fatigue for software developers. The observational study we conducted involved observing the work and sleep patterns of nine 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 them to assess their mental state and their performance as per the task. The study helped us evaluate ten research questions with respect to the discovered effects of fatigue. 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.

1.1 Initial Research Questions

In order to validate our understanding and to resolve the challenge of quantifying and measuring fatigue with respect to programming tasks, we set initial research guidelines for building a novel framework for to classifying effects of fatigue and relate it to the activities of a developer. First most, we wanted to establish how prevalent fatigue was perceived as a problem in industry:

- RQ1: *How severely and frequently does fatigue impact developer productivity?* If developers really do have concerns about fatigue affecting their work, that would be an indication of the underlying problem

Second, we wanted to establish an initial framework for characterizing the effects of fatigue:

- RQ2: *What are the factors that lead to mental fatigue and how does it affect a programmer's life?* To set some research guidelines for future researchers or tool-smiths to help developers understand their mental state, the fundamentals of fatigue should be understandable.

We conducted a study to answer these questions. We then validated these factors by conducting an observational study using our Eclipse (Java IDE) tool: *DevFatigue*, that provided empirical evidence of the effects of each factor of fatigue.

1.2 Thesis Statement and Contribution

Fatigue is a part of daily-life problem. In software industry we often see symptoms of mental fatigue in developers. *"We can discover the fundamental causes and effects of mental fatigue in developers and identify the predictors of fatigue based on a developer's interaction history".* The tools using these predictors can help developer detect and get rid of their mental fatigue state.

This thesis makes the following contributions:

- it presents results from a survey of 311 software developers, describing their experience with fatigue, their perception about fatigue, and how fatigue affects their work,
- it presents the results of an observational study of 9 professional software developers, supporting the results obtained from the survey study and assessing the features that could be applied in detecting fatigue in a developer, and
- it synthesizes the results of these two studies to suggest ways in which future researchers can support developers in detecting their fatigue state and sharing best practices.

CHAPTER

2

BACKGROUND

2.1 Defining Fatigue

Fatigue is a physiological state of reduced mental or physical capability [Muc06]. It results from excessive workload (both physical and mental), exhaustion, or sleep-loss. However, this definition is ambiguous as it is very broad and general. Fatigue (also called exhaustion, tiredness, and lethargy) is a complex phenomenon encompassing several physiological and psychological characteristics [Muc06]. Fatigue can be categorized according to the distinct effects it has on a person. Some of the categories are as follows:

- The inability to perform any physical activities at the level of one's normal activities is *physical/muscular fatigue*.
- *Mental fatigue* is defined as a state of weariness, with a feeling of boredom/saturation and decline in motivation [Aar99].
- Dull emotional responses is characterized as *Emotional Fatigue*.
- *Skills Fatigue* deals with the inability to perform a certain task at one's usual capacity.

Further, people are susceptible to fatigue that could potentially hamper their daily activities by affecting their ability to perform and thereby contributing to cognitive limitations that lead to undesirable results.

2.2 Cognitive Issues in Developers

Cognitive studies traditionally deal with the human mind and its processes. Concepts such as reasoning, perception, intelligence, learning, memory and emotion are studied in detail. Researchers have studied the cognitive issues responsible for several problems developers experience [PR12]. Often developers suffer from psychological disorders such as stress, insomnia, and exhaustion, which may be caused by underlying physical condition they are not aware of [Nan12]. They may make mistakes and therefore introduce bugs during software development. These mental errors are due to reduced cognitive functioning like distraction, low decision making power, diminished reasoning capabilities, and poor attention [Lar97].

Fatigue is considered as one the main agent of human errors. In industry, long and demanding tasks trigger mental fatigue [Aar99]. It is not uncommon for developers to be fatigued and worn out due to the high strain and many hours at work. Mental fatigue also has a negative effect on memory and cognitive functionalities that play a vital role in program construction and modeling [SM79]. Therefore, this poses the question of how mental fatigue affects programming tasks.

2.3 Mental Fatigue in Programming

Researchers have conducted several studies to understand the risk of reduced cognitive capability while driving [Bau08], nursing [Bar09] or performing other physical activities [Sai99] in industry. However, very little research is carried out in assessing the risk when programming. Smith and Miles conducted a study which analyses the effects of lunch on cognitive vigilance tasks [SM86], comparing the differences between participants with and without lunch. Another body of research considers mood as a factor affecting debugging performance, which also comes under the umbrella of psychological causes[Kha11]. Other than mood, stress, exhaustion, and sleepiness are some of the other factors, causing fatigue, to be considered in software industry. Literature reports that mental fatigue affects various different activities like reasoning, memory tasks, performance, decision-making and responsiveness [Bar09] [Sai99] [Win05] [MM96] [Kah70]. Fatigue, however, is generally described subjectively due to the variety of associated symptoms and causes. An industrial study [Sai99] quantifies a few subjective symptoms of fatigue in performing general tasks, relevant to the programming domain, shown in Table 2.1. Articles [Mak11] [Dan11] and blogs [Nan12] discuss the

Table 2.1 Subjective symptoms of fatigue.

Exhausted & Drowsy	Mental decline	Incongruity in body & nervous systems
Sleepiness	Nervousness	Physical Strength
Restlessness	Unwillingness	Pain in limbs
Feeling tired	Lack of focus	Strain in eyes
Sluggish	Absentmindedness	Dizziness
Lethargy	Anxiety	
	Weariness	

effect of fatigue on the efficacy of developers in tasks like program understanding, and decision making. Studies have shown possible relations between mental fatigue and specific programming tasks such as program construction, modeling and debugging [SM79]. 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 [FT03].

2.4 Fatigue and Effects on Performance

Performance is the process of accomplishing a given task computed against preset known standards of precision, completeness, cost, and pace. Performance can be calculated by monitoring the working patterns of a user and analyzing any change in them, as compared to other psychological effects of cognitive failures. For example, Pimenta et. al. [Pim13] [Pim14] have worked on monitoring and analyzing human performance with respect to computer related tasks and detailing the effects of fatigue on performance. Affective computing techniques have been used in detecting fatigue with respect to degradation in the performance. Overall, these techniques provide a useful but incomplete foundation about effects of mental fatigue in programming tasks.

Studies generally focus on the effect of fatigue on performance. Researchers have studied the effects of caffeine intake on alertness and performance [Smi93]. Coetzer and Richmond [CR07] performed an empirical analysis on working in teams and its relation to performance. *Performance is one of the known reasons, but can we be more specific about the effects of fatigue?* Our study aims to understand the fundamental reasons for performance degradation due to fatigue and ways to detect it in programming tasks. This will provide a broad spectrum for comprehending fatigue and developing new detection mechanisms. Surprisingly, in accordance with our literature survey, no study has been conducted to monitor the effect of fatigue on programming related tasks. We believe it requires an underlying understanding that can aid us in detecting a developer's state of mental fatigue and help cope with it.

CHAPTER

3

STUDY 1: SURVEY

To answer the research questions described in Chapter 1, we conducted a survey of software engineers seeking their experiences and views on fatigue.

3.1 Participants and Method

The survey [1] consisted of 11 questions grouped into categories: sleep, fatigue, work [Appendix A]. The questions about sleep were designed to get information regarding their sleep habits and circadian rhythm. Fatigue questions were asked to learn about their views on the fatigue in programming. We derived questions related to work to get information about their work habits and performance. They were intended to provide context for our interpretation. The survey was distributed with checkbox.io [2]. Participants entered a drawing for gift cards. We posted links to this survey on Reddit groups, Quora, in Computer Science Facebook groups, and emailed out to list servs that were directed towards software developers.

To answer **RQ1**, One of our survey questions was on a Likert scale from 1 to 5 asking developers to rate the severity and frequency of the effects of fatigue according to their personal experience.

[1] https://github.com/sarkarsaurabh27/fatigue/blob/master/study/survey.md
[2] http://checkbox.io/

To answer **RQ2**, both authors derived some factors that might cause mental fatigue: *Stress, Mood, Sleepiness, Physical work,* and *Non-willingness towards work,* based on existing literature [Sai99] [Aar99] [PR12]. We asked the respondents to choose all those causes which they have experienced. The question aimed to get a picture of, according to developers, the prime causes behind fatigue related to software development.

To identify the effects of mental fatigue, 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?* The responses to this question helped us define some factors affected by mental fatigue in programming tasks. To sort the factors, both authors randomly selected 30 responses in order to identify preliminary reasons. We repeated the exercise twice, once individually and the next time together. We compared the labels and finalized 6 labels. We used an online[3] open card sort to group the responses to this question into factors affecting programming tasks. Card sorting is a technique used to create mental models and define taxonomies from data [BZ14]. The inter-rater agreement scores among the authors for both the exercises are 21/30 & 26/30, respectively. Here, X/Y reads agreeing on X labels among Y cards. Primarily, the disagreement was between *Productivity* & *Performance* and *Mental Work Set* & *Motivation*. The confusion between Productivity and Performance was due to the overlap in the definitions. To help us sort better, we define **Performance** as the progress and speed of completing a task and **Productivity** as the completion of a task, mainly the quality of the output.

3.2 Results

We received responses from 311 participants. The age distribution of the respondents lies mostly between 17 and 74. Around 50% of the participants are between the age of 20 to 35. The responses of our survey show developers' views towards fatigue. They helped us answer the research questions:

RQ1: *How severely and frequently does fatigue impact programmer productivity?* From our survey, 66% of the respondents thought it has a high severity. 59% respondents stated that it occurs to them more often than not. The responses we gathered, shown in Fig. 3.1, motivated us to delve into the problem of mental fatigue of developers in software industry.

RQ2: *What are the factors that lead to mental fatigue state and how does it affect a programmer's life?* Both authors ranked the causes according to the responses, shown in Fig. 3.2. *Stress* and *sleepiness* are the highest ranked factors, which validates our definition of mental fatigue in programming tasks', that triggers from sleepless nights and day-long work. Physical activities often play a role in

[3] https://conceptcodify.com/

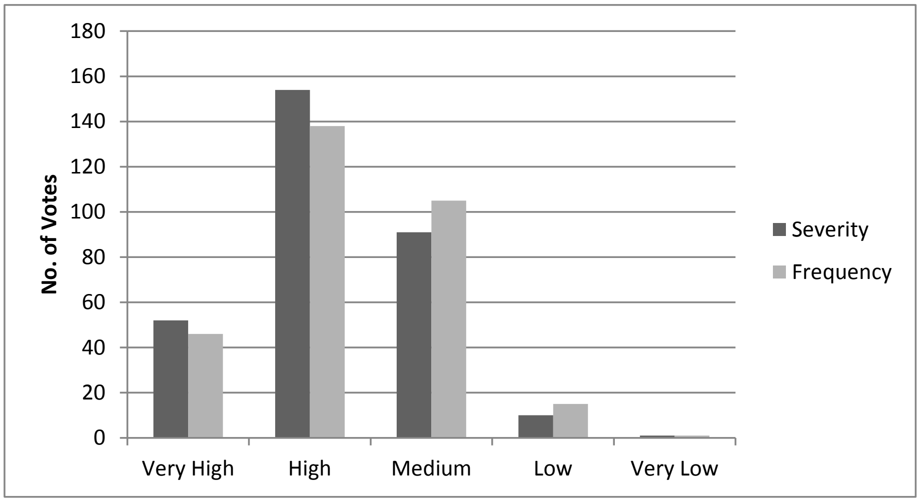


Figure 3.1 Voting results about the severity & frequency of fatigue by the respondents.

triggering exhaustion as well. We found *mood* among the selected causes, which is already considered having an effect on programmer's debugging performance [Kha11]. Respondents agreed that exhaustion affects their work. We interpreted their experiences with fatigue and discovered some factors that affect programming. The effects of fatigue on programming tasks are reflected in Section 3.3.

3.3 Effects of Fatigue

Several implications allowed us to speculate on the way developers behave when they are in the fatigue state. Responses from our survey helped us gather data about fatigue and investigate possible detection and alleviation mechanisms. We distilled the reported experiences into six factors, using card sort, shown in Fig. 3.3. Factors are ordered from the highest to lowest frequency of labels. Frequency is mentioned in-line with each factor. A response from the survey that best depicts the factor is listed after each description.

- **Performance:** [28.42%] Performance is the most studied factor for any type of task, whether it is a computer related task [Pim13] [Pim14] or a physical one [Sai99]. Performance is calculated with respect to the progress of a task and speed of the progress. Suboptimal solutions and having less reasoning capabilities contribute to degradation of performance. Responses in this factor also acknowledged that they lose clarity to think and they often need to redo the

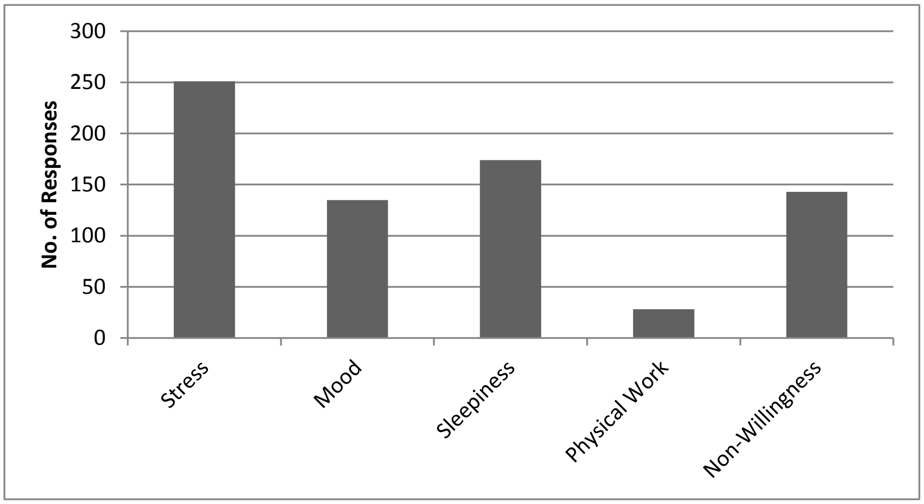


Figure 3.2 Causes of mental fatigue selected by the respondents, supported by their experiences.

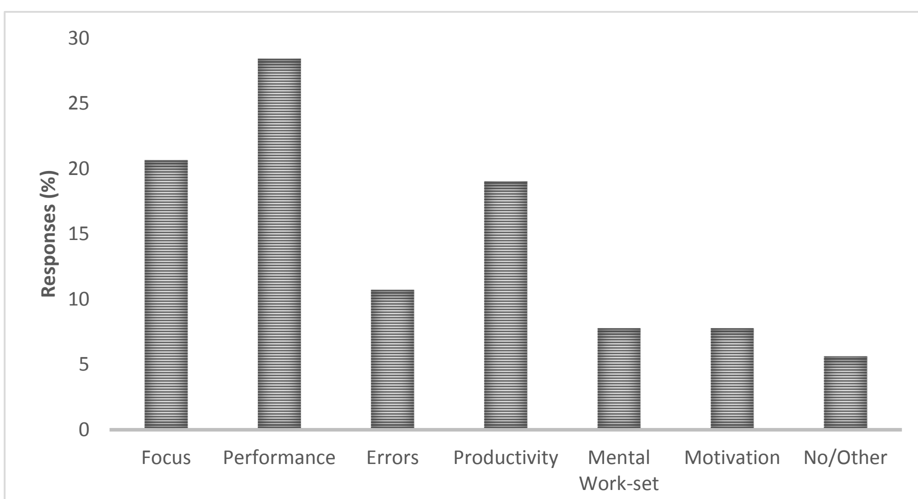


Figure 3.3 Distribution of the factors leading to mental fatigue, discovered by the survey responses of 311 developers.

work.

"When I am tired *I cannot come up with solutions as quickly, or sometimes not at all...*"

- **Focus:** [20.64%] Responses mostly presented poor attention and absentmindedness. Respondents were concerned about being easily distracted and less productive. Impaired concentration leads to lack of regard for details, resulting in lower quality.

"When I am tired I find myself drifting from tasks that require a lot of *focus* (mental effort), or making programming choices that are inefficient..."

- **Productivity:** [19.03%] Productivity [Par14] is defined as the extent of the performance with respect to the system goals and completion of those goals. Responses have also acknowledged an intersection between performance and productivity. Apart from poor reasoning, it also affects quality and creativity.

"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..."

- **Errors:** [10.72%] When the mind is tired, developers tend to make mistakes and sometimes forget to stick to standards. Distraction and lack of focus is also a contributor in introducing bugs. Mostly the responses pointed out making logical errors and silly mistakes.

"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...*"

- **Mental Work-set:** [7.77%] Mental Work-set is defined as a set of concurrent mental tasks on a programmer's mind at any given time. In mental fatigue state, a developer's mind looses the ability to focus on more complex tasks and ability to coordinate concurrent activities is diminished. Developers tend to escape from complex tasks and focus on unnecessary trivial tasks.

"I *cannot keep large coding context in my memory*. I zone out all the time. I feel demotivated. *Similar to coding while drunk...*"

- **Motivation:** [7.77%] People have their own reasons to get motivated. Sleepiness induces unwillingness to work. Responses acknowledged that without motivation to work, developers move towards delaying the work, in fact, they are unwilling to even start.

> "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'…"

3.4 Threats to Validity

Although our survey study provides data on fatigue and its detrimental effects on programmers, there are several threats to validity that should be considered when interpreting our results. First of all, survey population may not generalize to all developers. To mitigate against this threat, the survey was posted on various groups assuming that those groups are mostly accessed by professional software developers but this may. As all the fields of the responses were not mandatory, so some responses were blank. These type of threats are mostly common in case of online surveys. Finally, the interpretation and analysis of the factors discovered from the survey responses was performed by one author using open card sorting technique. To avoid observer bias, initial card sorting was performed by both the authors and was taken further for more analysis only after achieving a satisfying inter-rate agreement score.

CHAPTER

4

STUDY 2: OBSERVATION

The survey results raised many questions. For instance, is the perception of fatigue clear and common to all software developers? How, when and what kind of tasks developers opt when they are fatigued? How can we justify developers' response by their activities? To investigate the usefulness of identifying fatigue during programming tasks and recognizing its impact on performance, we evaluated the outcome of the survey study by conducting some other empirical studies on developers. Along with monitoring the developers' work activities, the observation study included a sleep test for the participant and surveys about their work. Specifically, we aimed to be able to identify and validate the observed factors from their interaction history.

4.1 Research Questions based on the Effects of Fatigue

Many claims have been made by researchers concerning "operator fatigue" of workers engaging in long running tasks. One claim was that key-down time decrease with respect to approaching fatigue state [Pim13] [Pim14]. Ideally, that is expected to occur over the period of time in a day because programmers are less likely to be involved in very repetitive tasks. Instead, we examine differences in the overall key usage of a fatigued day in comparison to a normal day. Our research question centers around how fatigue affects the **performance** in terms of cognitive skills related to keyboard interaction and mouse usage in software development process? Even though those researchers have examined how decrease in cognitive skills in terms of keyboard and mouse usage would reflect human performance of interaction with a computer, the nature of effect of fatigue on programming tasks is not clear.

- RQ3: *Could mental fatigue be detected through the handling of keyboard and mouse usage data in developers?*

Software developers often work on multiple projects at the same time. While working on a single task during program comprehension, developers flip between contexts. Researchers have suggested ways to improve developers' ability to switch and recover the mental state associated with a task [PG06]. Our next research question focuses on how does fatigue affect a developer's actions while interacting with a task or how does it affects developer's **focus**?

- RQ4: *Would there be any effect of fatigue on the user interaction with respect to the code context?*

Researchers have found various methods to evaluate the factors affecting **productivity** [TN04]. A significant factor for the software industry would be the quality of their products. Quality could be measured in number of ways. We measured quality in terms of adherence of coding standards, thus leads to our next research question.

- RQ5: *Would we see any drop in the quality of the code?*

The following research question revolves around the most common effect of fatigue. Along with the effect on the interaction with the computer, researchers have found increase in the number of **errors** per key [Pim13] [Pim14]. Is there any effect on the numbers of errors made by the developers with respect to coding, when fatigued?

- RQ6: *Would we notice increase in the numbers of errors and corrections per key?*

Mental Work-set has been discovered as a factor of fatigue by the survey responses. To confirm the consistency of the survey responses and to prove the phenomenon, we need to answer two RQs:

- RQ7: *Would we see smaller amount of files?*
- RQ8: *Would we detect more "rote" tasks, like systematic manipulation of the code?*

Survey respondents stated that when they feel fatigued, they tend to work less and sometimes its hard to even start working in the first place. We believe, in low **motivation**, keys and mouse usage would depict different behaviors in terms of interaction with the computer. We want to verify this assertion by asking the following research question about the user activity:

- RQ9: *Would there be any difference between the keyboard interaction and mouse usage?*

Moreover, a research question about the time spent on a task by a developer would also help depicting the survey responses.

- RQ10: *Does fatigue affect the time spent by developers on IDEs?*

4.2 Participants and Method

Nine professional software developers participated in the observational study. Some participants were recruited from online free-lancing websites[1][2], on the basis of their daily work routine and expertise with the Eclipse IDE. We selected other participants through personal contacts. If they were willing, we sent them a consent form and pre-study questionnaire. On an average, the participants have an industrial experience of 3 to 4 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' demographic information was collected on the time of recruitment and was very diverse in order to cover a large working culture across the globe. The participants were not given any specific task to perform, they were asked to work on their own task which was monitored by our tool (Eclipse plug-in).

We implemented a plug-in for the Eclipse IDE. DevFatigue[3] is an activity tracking plug-in for Eclipse. It is an extension of Rabbit[4]. Like Rabbit, it works in the background with Eclipse and tracks all the activities you perform. It only tracks the actions when Eclipse is active. And logs the data in XML (human readable) format at specific location.

Rabbit features can be referred to on its official site[4]. Additional features in DevFatigue include:

- User Activity - The typing speed (key board usage) of the user with respect to a time period.
- Focus Events - The activities related to keys and mouse usage, like Key Up, Key Down, Mouse Clicks, Mouse Velocity, etc. with specific to time period.
- Project Events - Information regarding the projects like 'imports', and commands used with respect to time period.
- Navigation Events - Information regarding the files, line numbers and methods traversed by the developer with the time-stamp.

[1] freelancer.com/
[2] elance.com/
[3] http://www4.ncsu.edu/ ssarkar4/fatigue/eclipse/updatesite/
[4] https://code.google.com/p/rabbit-eclipse/

- Self Assessment - The plug-in includes a daily and a session survey.

The architecture of the proposed framework is the extension of the frameworks used in some of the previous studies [Pim14]. The framework collects stylometrics, specifically keystroke and mouse dynamics, that will help in detection and classification of degradation in performance, which can be an effect of fatigue. We are trying to collect the data in a non-intrusive and dynamic way and the indicators of mental fatigue recorded by DevFatigue are:

1. Keydown Time: time spent between two consecutive key down and key up events.
2. Errors per Key Pressed: number of times a correction is made per key pressed.
3. Mouse Velocity: velocity of the cursor.
4. Mouse Acceleration: acceleration of the mouse.
5. Time between Keys: time spent between two consecutive key up and key down events.
6. Time between Clicks: time spent between two consecutive mouse up and mouse down events.

All the above mentioned indicators have been proved useful in previous studies both statistically and empirically [Pim13] [Pim14]. These indicators are collected in a specific time span and provide us information we need to analyze the working patterns of the users and to infer whether the programmer is fatigued or not.

We define Productivity in terms of the quality of the work done by a developer. To calculate the quality we analyzed the project base using a static code analysis tool - *Checkstyle*. Checkstyle is a development tool to help programmers write better Java Code by enforcing coding standards [Bur07]. Our analysis supported sun code conventions[5] for the standards.

The pre-study questionnaire was a sleep test to recognize and detect symptoms of any sleep deprivation related to sleeping disorders. The sleep test is developed by Dr. Russell Rosenberg and is easily available online[6]. We added a question about the developer's daily sleep routine. The sleep test helped us categorizing the participants and find out any difference in their work patterns according to their test results. The sleep test is shared in Appendix B.1.

Along with the sleep test, the study included a daily and session surveys. As software developers usually have busy days in software development activities including coding, reviews, debugging and mainly team meetings. There was a challenge of coming up with a set of questions that every participant would feel comfortable to answer and which would not be an overhead on their existing

[5] http://checkstyle.sourceforge.net/
[6] https://www.ohiohealth.com/sleeptest/

tasks. Studies have shown that tool-smiths face continuous challenge of getting developers to adopt new tools [Sni14]. The rationale behind the survey is to collect information to tag the user activity data collected by DevFatigue. We wanted to tag the data to differentiate between the working patterns of the developer depending on his/her mental state. We expected the developers to respond to the survey honestly and to the best of their knowledge. DevFatigue Self Assessment Survey is to get the participants self-assess their mental state with respect to performance, fatigue, frustration, etc. It also asks questions about their project and their work/sleep patterns. This is done in two forms:

1. Daily - 15 questions are asked about the overall task performed over the day. The daily survey was in a form of a pop-up (Eclipse View) which can be invoked on any selected time in the whole day.

2. Session - A single question is asked about their fatigue(tiredness) level in the last session. The session survey gets invoked on every user session depending on a break of 15-20 minutes.

The survey included questions about their daily activity based on some themes:

- Sleep routine
- Project Information
- Lifestyle
- NASA-TLX

The survey questions are shown in Appendix B.2. The questions of the survey were mostly inspired by the NASA-TLX (Task Load Index) [HS88] study. NASA-TLX[7] is a subjective workload assessment tool. NASA-TLX allows users to perform subjective workload assessments on operator(s) working with various human-machine systems. It included some text based questions and some true-false based questions. The questions 7-14, based on NASA-TLX, were on a Likert Scale of 0-10 and the responses were used to score their mental state. Similarly, in the session survey, the question about their fatigue level was also on the Likert Scale of 0-10.

4.3 Results and Analysis

To investigate the factors discovered from the survey results in more depth, we conducted an observational study of software developers at work including daily and session surveys. To focus

[7] http://humansystems.arc.nasa.gov/groups/tlx/

Table 4.1 Data collected by 56 logs generated, by DevFatigue, for 9 participants.

User	Total Days	Low Fatigue Days	Medium Fatigue Days	High Fatigue Days	Avg. TLX Score (Out of 80)	Total User Activities	Total Time Worked on IDE (Mins.)
1	8	3	5	0	33.12	9855	863.18
2	8	5	2	1	25.12	13822	346.95
3	8	2	3	3	47.25	2110	134.23
4	8	8	0	0	24.75	21584	662.76
5	4	2	1	1	30	3982	113.33
6	6	3	2	1	37.25	3079	233.75
7	3	3	0	0	41.5	1546	54.76
8	6	3	2	1	36	6566	743.11
9	5	2	2	1	24.5	5708	155.16
9	56	31	17	8	33.27	68252	3307.23

the observational study, we asked more research questions and answered them with the help of the observation study results. All the participants consented for the study and filled out the the sleep test. The sleep test was used to detect sleep disorders like *sleep apnea, insomnia, narcolepsy, gastroesophageal reflux and periodic limb movement disorder*. The results showed that, among the nine, one of the participant had symptoms of 'Sleep Apnea', two participants had 'Insomnia' and one participant had 'Periodic Limb Movement' disorder, which none of them were aware of. The question about their sleep routine showed that all participants have a similar circadian rhythm, hence comparable data. Table 4.1 shows the various attributes of the collected data for each participant from 56 days logs generated by DevFatigue. We observed that activities and time worked is comparable to the task-load index score.

We intended to come up with a focus curve that can show exhibit working patterns of an developer and help us determine a model based on the data collected to understand the effects of fatigue depending on the developer's activities. In our analysis, we defined *Energy* as the number of activities performed in a time span. The graph in Figure 4.1 shows the working patterns of an developer over a period of days comparing different fatigue levels. The corresponding curves are specified by different colors. The graph shows a common pattern that we observed across all the participants. We observed that there is a significant difference in the working patterns of the developer for each day and it motivated us to delve into more low level analysis to find out the various factors and to help answer our research questions. The patterns also showed that there was a stark difference in

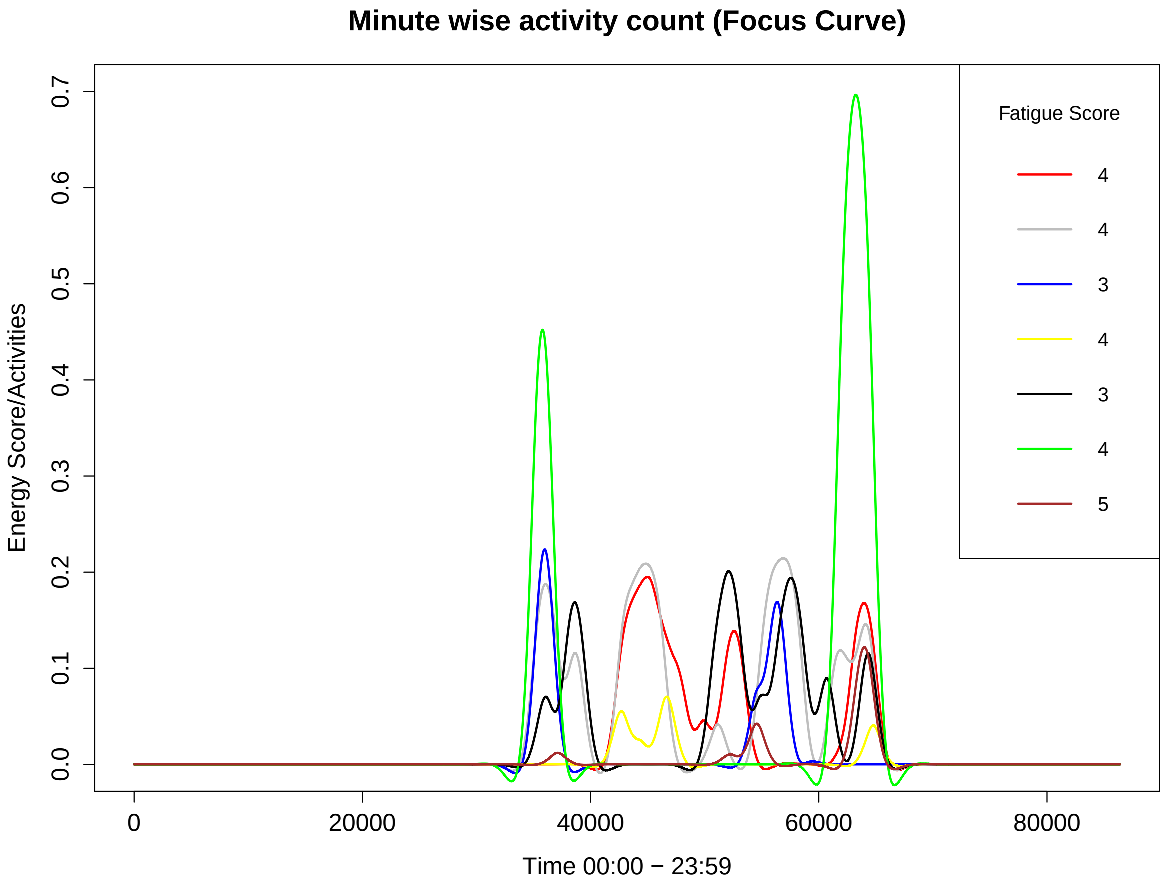


Figure 4.1 A representative example of focus curves comparing energy level on different days with respect to the fatigue level score.

other workers previously studied in fatigue studies, where the work pattern would show a steady decline of activity as the day progressed.

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 [Leh05] 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. Besides correlation coefficient, we calculated student's t-test (two tailed) score between individual fatigue levels for some of the features as well. The analysis of each of the discovered factors can be found next.

4.3.1 Performance

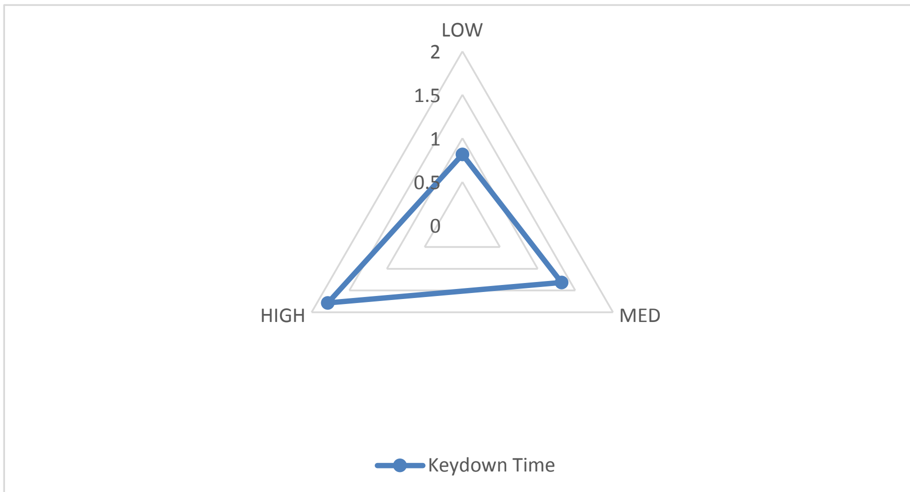
RQ3 relates to detection of fatigue through keyboard interaction and mouse usage data. We expect that there would be a decrease in the cognitive skills which could be represented by general patterns of slower mouse and keyboard interactions. This claim has been already proven by other studies [Pim13] [Pim14], but not in software development related tasks specifically.

To answer RQ3, we examined the indicators of mental fatigue as proven useful by previous studies [Pim13] [Pim14].

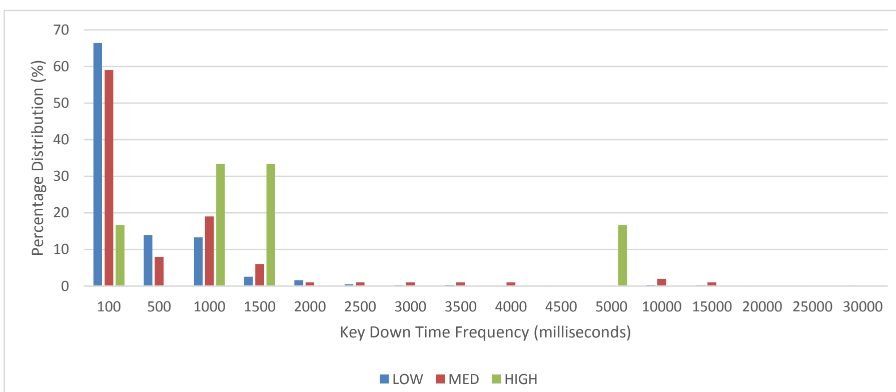
- **Key-down Time**: Our hypothesis states that the key-down times will be more in a fatigued day as compared to in a normal day. Figure 4.2a shows the aggregated results of all the participants calculated by taking the mean of the weighted average of the frequencies of the key-down times for respected fatigue levels. The figure represents a radius graph where the distance from the center depicts the average key-down time. Each corner of the graph denotes different level of fatigue. It is clearly indicated in the graph that the key-down time is comparatively more in case of a higher fatigue level.

  For instance, let's just consider the percentage distribution of the weighted average of the frequencies for each level which is shown in Fig. 4.2b. The most common pattern we observed that in a low fatigue level the low frequencies covered around 97% of all the key-down times and rest is distributed among the other frequencies. In case of medium fatigue level, the low key-down frequencies cover only 90% of the distribution and in case of high fatigue level, it covers around 80% of the distribution. We categorized low key-down frequencies as 1 to 1500 milliseconds. 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 0.999784 which denotes a strong correlation measure. This test examines the difference in the pattern of key-down times for different fatigue levels and the results support our hypothesis that we have larger key-down times for a higher fatigue level.

- **Time between Keys**: We have a similar hypothesis for time between keys, which states that the time between keys will be more in a fatigued day as compared to in a normal day. Figure 4.3a shows the aggregated results of all the participants calculated by taking the mean of the weighted average of the frequency of the time between keys for respected fatigue levels. The figure represents a radius graph where the distance from the center depicts the average time between keys. Each corner of the graph denotes the different level of fatigue. It is clearly



(a)



(b)

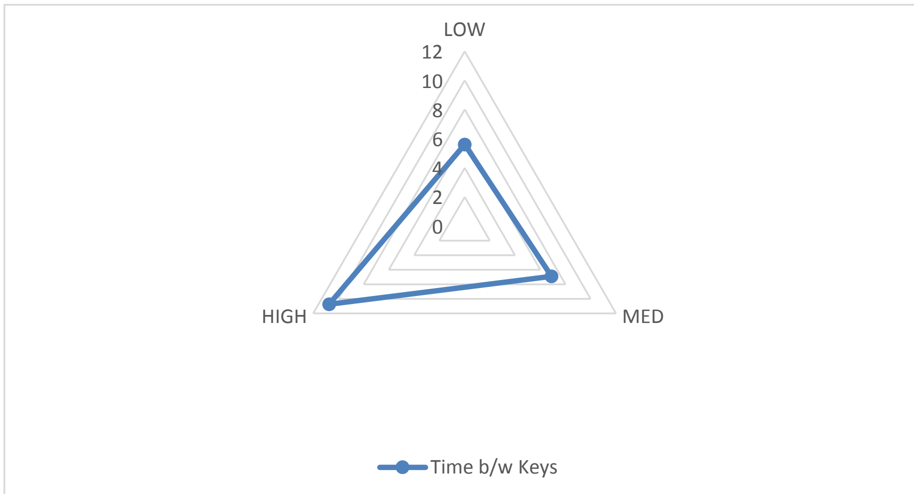
Figure 4.2 a) A radius graph b) A column bar comparing the Key-down Time of all the participants together to the corresponding fatigue level: in high fatigue level, developers tend to code slower.

indicated in the graph that the time between keys is comparatively more in case of a higher fatigue level.

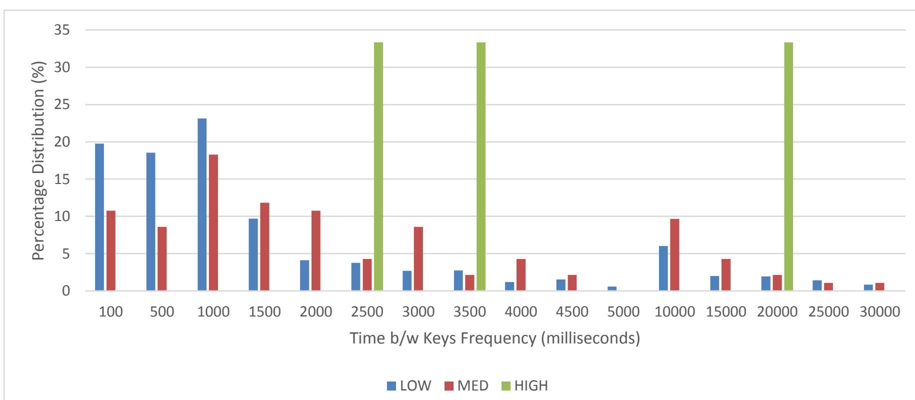
Figure 4.3b depicts an example, where we have the percentage distribution of the weighted average of the frequencies for each fatigue level. The most common pattern we observed that in a low fatigue level the short delays covered around 75% of all the time between keys and rest is distributed among the other delays. In case of medium fatigue level, the short delays cover only 60% of the distribution and in case of high fatigue level, it actually does not include any short delays of the distribution. We categorized short delays as 1 to 2500 milliseconds among the frequency distribution. In high fatigue level, the distribution is among the long delays. We also calculated the Spearman's rank correlation coefficient comparing the time between keys with respect to the fatigue score associated with them. We found out the coefficient as 0.960758 which denotes a significant contribution. This test examines the difference in the pattern of time between keys for different fatigue levels and the results support our hypothesis that we have larger time between keys for a higher fatigue level.

- **Time between Clicks**: Mouse usage also contributes to the detection of fatigue. We expect that the time between clicks will be more in a fatigued day as compared to in a normal day which is shown by Fig. 4.4a. The graph shows the aggregated results of all the participants calculated by taking the mean of the weighted average of the frequency of the time between clicks for respected fatigue levels and is represented by a radius graph. Each corner of the graph denotes the different level of fatigue. It is clearly indicated in the graph that the time between clicks is comparatively more in case of a higher fatigue level.

  Figure 4.4b depicts an example, where we have the percentage distribution of the weighted average of the frequencies for each fatigue level. The first thing which we observed that there were more mouse events (clicks, scrolls, etc.) in a higher fatigued level comparatively. The other observation was that, in high fatigue level the delays were in the time cluster from 4500 to 30000 milliseconds mostly, whereas in lower fatigue state session it was scattered. We can conclude that in higher fatigue level user demonstrates a sluggish mouse usage comparatively and it remains constant through out the session and usage of mouse is higher compared to that of keys in higher fatigue level. We also calculated the Spearman's rank correlation coefficient comparing the time between keys with respect to the fatigue score associated with them. We found out the coefficient as 0.99451 which illustrates the significant increase in the time between clicks with respected to the fatigue score. This test examines the difference in the pattern of time between clicks for different fatigue levels and the results support our hypothesis that we have larger time between clicks for a higher fatigue level.

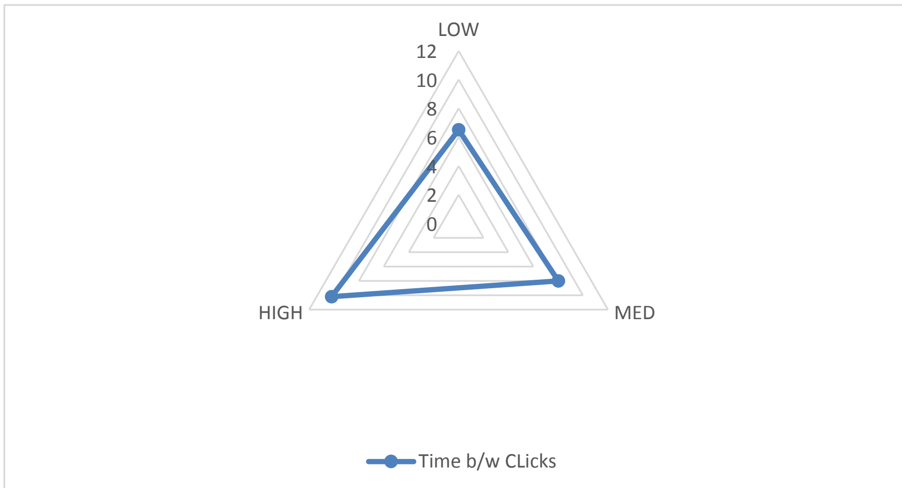


(a)

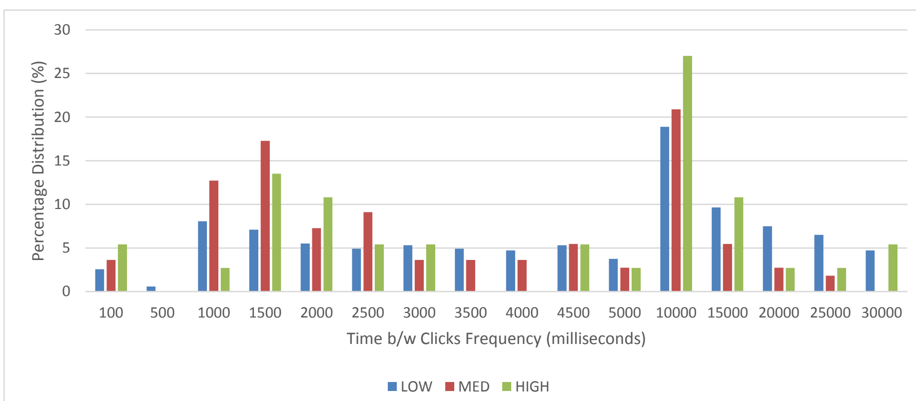


(b)

Figure 4.3 a) A radius graph b) A column bar comparing the Time between Keys of all the participants together to the corresponding fatigue level: in high fatigue level, developers tend to code slower.



(a)



(b)

Figure 4.4 a) A radius graph b) A column bar comparing the Time between Clicks of all the participants together to the corresponding fatigue level: in high fatigue level, developers tend to click mouse slower.

- **Mouse Velocity**: Slow mouse movement denotes a sluggish behavior in terms of performance. Our claim for mouse velocity states that there will be a decrease in the mouse velocity with respect to an increase in the fatigue level. Figure 4.5a depicts a comparison of the mouse velocity with the fatigue level. The data is an aggregation of all the participants working pattern over a period of 7 days. Each corner of the radius graph denotes different level of fatigue. It is clearly indicated in the graph that the mouse velocity is comparatively less in case of a higher fatigue level.

  To observe some more analysis results, we represent the percentage distribution of the frequencies of mouse velocity in Fig. 4.5b. The Spearman's correlation coefficient is -0.78762 which denotes a significant decrease in mouse velocity with increase in the fatigue level, which supports our claim.

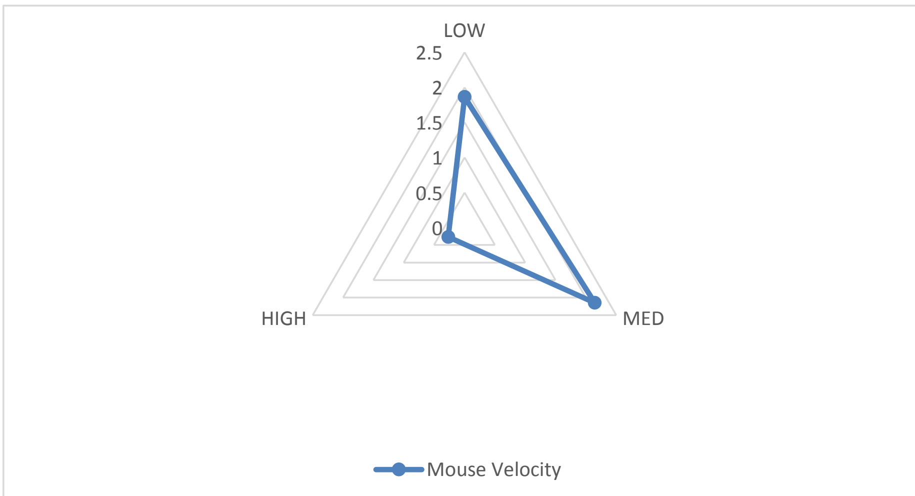
- **Mouse Acceleration**: Similarly our hypothesis for mouse acceleration is that there will be a decrease in the mouse acceleration with respect to an increase in the fatigue level. Figure 4.6a depicts a comparison of the mouse acceleration with the fatigue level. The data is an aggregation of all the participants working pattern over a period of 7 days. Each corner of the radius graph denotes different level of fatigue. It is clearly indicated in the graph that the mouse acceleration is comparatively less in case of a higher fatigue level.

  To observe some more analysis results, we represent the percentage distribution of the frequencies of mouse acceleration in Fig. 4.6b. The Spearman's correlation coefficient is -0.62234 which denotes a significant decrease in mouse acceleration with increase in the fatigue level, which supports our hypothesis.

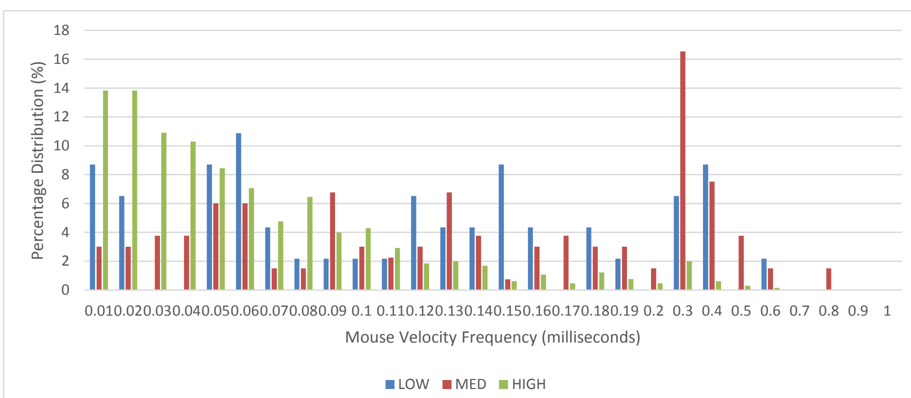
We analyzed the above indicators to address the question as presented in RQ3. We therefore conclude that keyboard and mouse interaction data place a key role in evaluating the performance in a programming task. We found out that there is a common pattern for each of the indicators that can help us detect fatigue in developers.

4.3.2 Focus

Since focus plays a crucial role in software development related tasks, we investigated how fatigue affects focus to answer RQ4. When a developer interacting with a code context during program comprehension, he should be focused on the task. In terms of software development process, it can be measured how much time is the developer spending on a particular file? What pattern does the developer follow to traverse through the files and methods? Does his/her interaction involves more navigation comparatively?

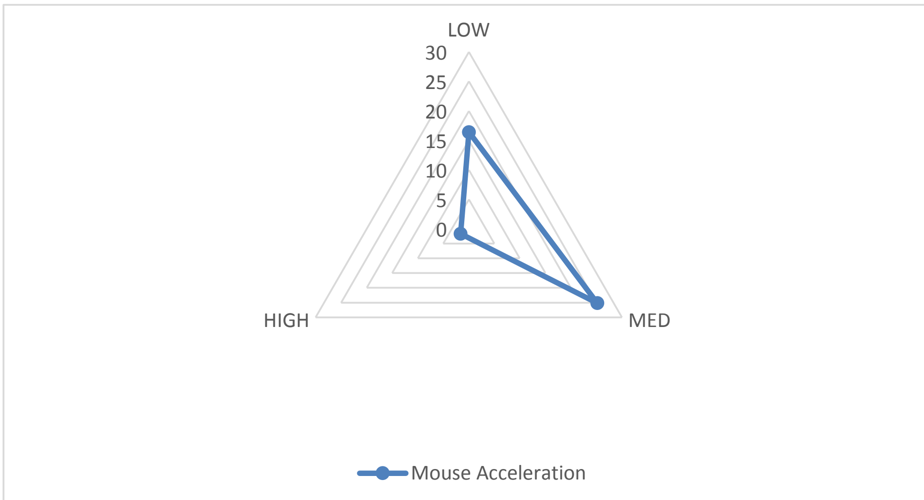


(a)

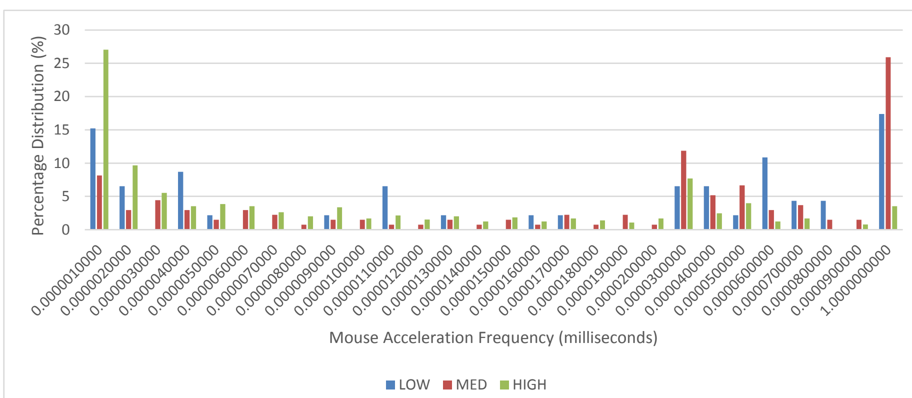


(b)

Figure 4.5 a) A radius graph b) A column bar comparing the Mouse Velocity of all the participants together to the corresponding fatigue level: in high fatigue level, developers tend to move mouse slower.



(a)



(b)

Figure 4.6 a) A radius graph b) A column bar comparing the Mouse Acceleration of all the participants together to the corresponding fatigue level: in high fatigue level, developers tend to move mouse slower.

To address these questions as presented in RQ4, we considered three features:

- **File Transition**: We expect that when a developer is more focused, he/she spends more time on a file. Figure 4.7 shows the average time per file spent in each fatigue level. We observed that with increase in the fatigue level there is a decrease in the average time per file. This test shows an indirect relation between focus and fatigue. The calculated Spearman's correlation coefficient is -0.91452 which supports our hypothesis that in case of fatigue a developer would spend less time on each files, hence less focused. Moreover, we calculated the t-test score and found that the data sets for each fatigue level is different from each other with a t-test score of: Low vs High = 0.47856, Low vs Med = 0.91302, and Med vs High = 0.50068. It does not denote a significance difference but considerable for the amount of data we have.

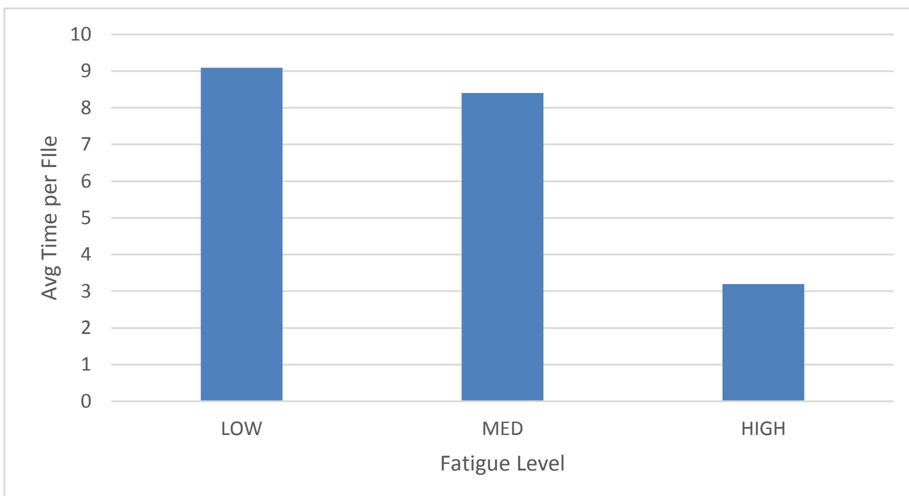


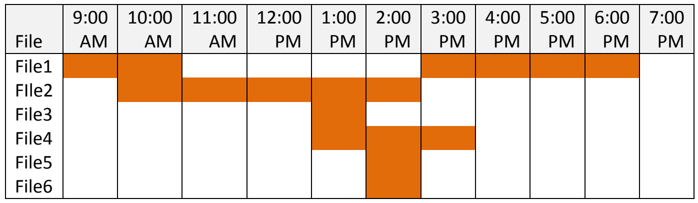
Figure 4.7 Bar plot comparing Average Time per File to the corresponding fatigue level: in high fatigue level, developers focuses on files for short periods.

If a developer is spending less time on each file and mostly navigating through the code base, we expect a difference in the file navigation pattern for each fatigue level. We assumed that in case of fatigue, developer would be unable to focus on tasks involving complex code. But, we found out that a developer traverses more files when he/she is in a higher fatigue level. This conflict in our assumption supports the above hypothesis of spending less time per file. The difference in the file navigation pattern can be observed in an example by taking three representative logs for each fatigue level. Low fatigue level is shown in Fig. 4.8a, medium fatigue level is shown in Fig. 4.8b and high fatigue level is shown in Fig. 4.8c. All the three examples are for an equal time span, i.e. 9:00 AM - 7:00 PM for a day. We observed that in the time span the developer is traversing more files in a higher fatigue level. In Fig. 4.8a, we notice that only 10:00 AM and 2:00 PM has multiple files and has a maximum of 4 files per hour rate. In contrast, in Fig. 4.8b, the maximum rate is 8 files per hour and in Fig. 4.8c, the maximum rate is 11 files per hour. It is clearly noticeable that there is less focus in a high fatigued day, as claimed.

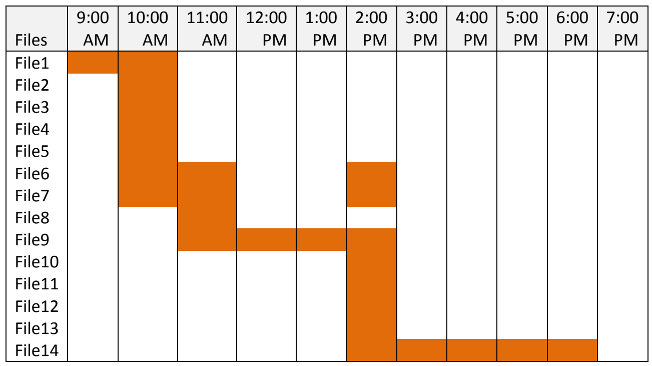
- **Navigation**: We can measure focus in terms of navigation pattern as well. We claimed that with increase in fatigue level their will be an increase in navigation as well. It is similar to the average time spent on a file, in addition it also calculates the navigation with in a file. We considered the navigation using key arrows, mouse clicks, mouse scrolls and the navigation functionality provided by the IDEs. We examined that there is an increase in both the cases with an increase in fatigue level. In fact, the hypothesis is supported by statistical analysis using Spearman's correlation coefficient giving a score of 0.949245 and 0.295376 for key/mouse navigation and command navigation, respectively. Figure 4.9 depicts how the data backs our hypothesis. It can be noted that the key navigation rate is increasing with increase in fatigue level and similarly, for system commands, high fatigue level has a higher rate of 0.035433 than the rate of 0.02371 in medium fatigue level. Developers tend to traverse more through the code base than actually coding, when they are fatigued.

- **Method Transition**: During program comprehension, transition is the change from one location to another. Along with file traversal, transition include method traversal as well. Researchers have already identified several types of method transitions like inter-class transition, intra-class transition, recovery transition and exploration transition [PG06]. From file transition, we found out that in high fatigue level developers do not spend time on any particular file for a long time and they have an indefinite pattern reflecting low focus. Similarly, we expect that in low fatigue state developers would focus more on limited number of methods than just navigating through them. We analyzed method transition by taking the frequency of each method traversed and plotting the distribution. For example, let us represent each method with a unique keyword, then the transition can be represented as Fig. 4.10. We took 3 representative data sets for each of the fatigue level, respectively and examined their frequencies, shown in Table 4.2.

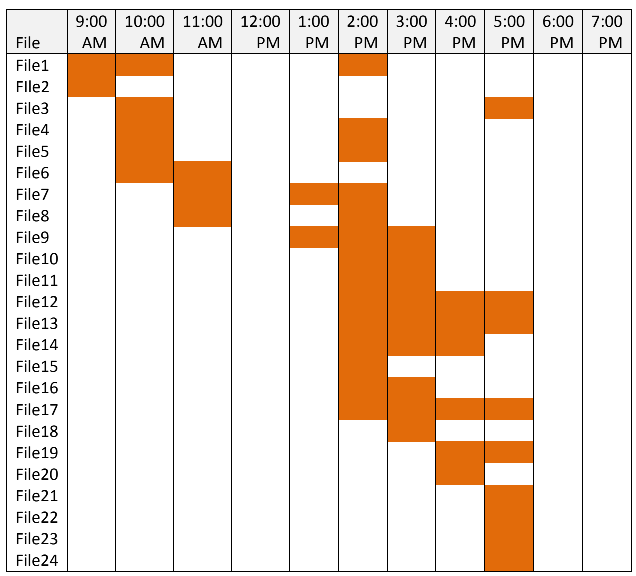
  Same has been represented in Figure 4.11. In the graph, we observe that the slope of the low fatigue level line is more than that of medium and high fatigue level line. Moreover,



(a)



(b)



(c)

Figure 4.8 An representative example: Matrices showing the file traversal pattern of a participant on a random day. a) for a low fatigue level day, b) for a medium fatigue level day, and c) for a high fatigue level day.

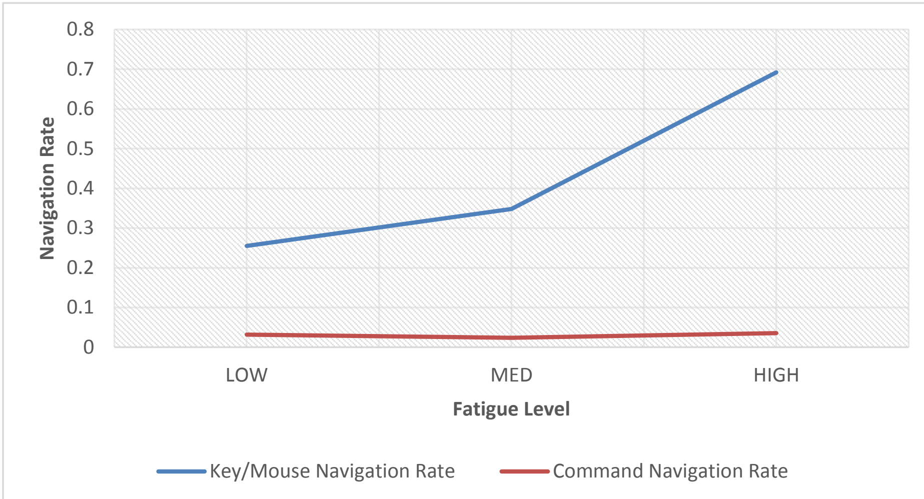


Figure 4.9 Line chart comparing the key/mouse navigation rate and command navigation rate to the corresponding fatigue level: in high fatigue level, developers tend browse more and code less.

Table 4.2 Example of method transition frequency and corresponding fatigue levels.

Low Fatigue Level		Medium Fatigue Level		High Fatigue Level	
A	1	A	4	A	4
B	5	B	1	B	1
C	17	C	7	C	1
D	1	D	1	D	3
E	1	E	2	E	2
F	3	F	1	F	1
G	3	G	1	G	3
H	1	H	1	H	1
I	14	I	1		
J	1	J	5		
K	1	K	4		
L	2	L	1		
M	1				
N	1				

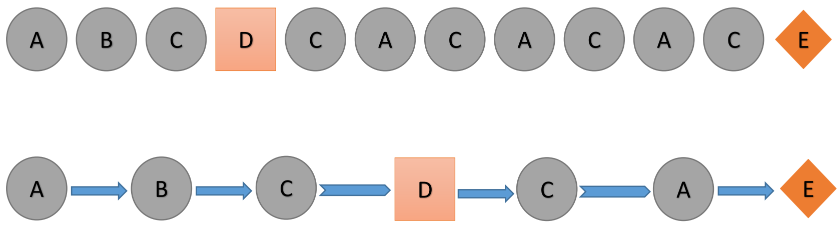


Figure 4.10 Example of method transition.

we examined, do these individual data sets fit Pareto Distribution[8]? The distribution with probability density function and distribution function defined over a period of time is Pareto Distribution. Pareto distribution is used to describe conditions in which an equilibrium is found in the distribution of the "small" to the "large". For example, in describing the distribution of wealth, it is sometimes expressed more simply as the Pareto principle or the "80-20 rule" which says that 20% of the population controls 80% of the wealth. Similarly, for our problem, we have the distribution for many methods but few focused on. We found out that the shape parameter α of each line (fatigue level) is 1.40392, 1.709138, 1.870616 respectively. We can observe from the graph that the data with low fatigue level is focusing more on less number of methods where as in high fatigue level day the frequencies (focus) of all the methods are somewhat same.

To answer RQ4, we examined the above features and found out that fatigue affects focus. There is a difference in the user interaction in case of fatigue. We found out that the above features can help us build a model to detect fatigue in a developer.

4.3.3 Productivity

RQ5 relates to how would fatigue affect productivity. Researchers have found that developers perceive productivity as completion of successful tasks without much interruption [Mey14]. We measure productivity in terms of **code quality** and our hypothesis states that there will be a fall in the code quality with increase in the fatigue level. We used Checkstyle, a static analysis tool which enforces coding standards, to evaluate the code quality. Checkstyle generates results in terms of errors in every file. We calculated code quality considering the change in the code analysis errors in a project for each day. We set the baseline as the initial error for the day from the previous day and calculated the difference at discrete time stamps. The error value for the whole day would be the mean of all

[8] http://en.wikipedia.org/wiki/Pareto_distribution

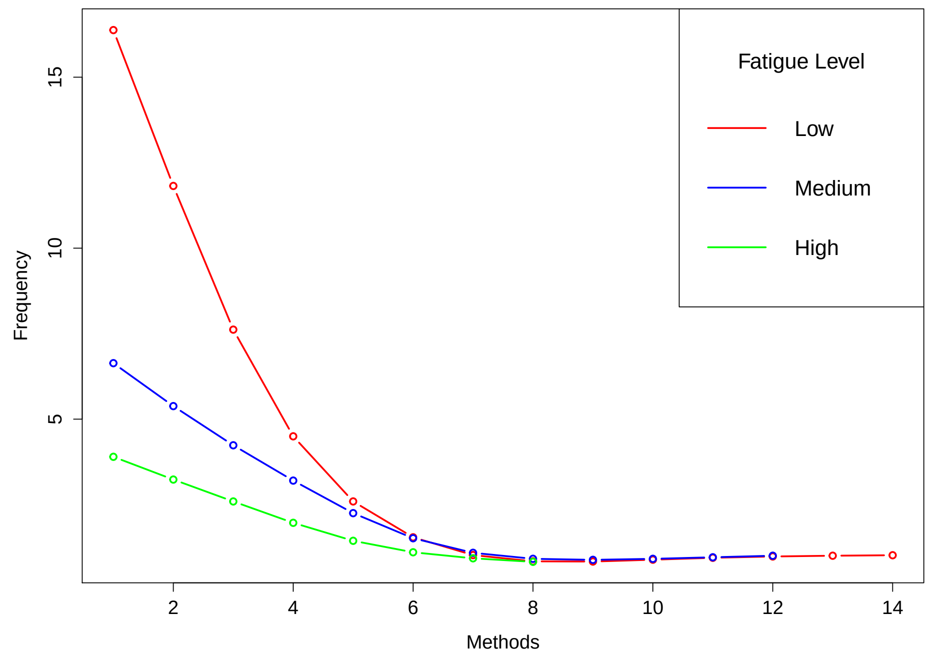


Figure 4.11 Line Chart comparing the method transition pattern for respective fatigue levels: high slope more focused.

the differences. We aggregated the whole data for each user and categorized it into three fatigue levels as stated earlier. Figure 4.12 represents a radius graph depicting the relation between fatigue and code quality. We observe that the code quality is higher when the fatigue level is low. The same hypothesis is supported by the Spearman's correlation coefficient, that is, -0.92984 which represents a significant decrease in code quality with increase in fatigue level.

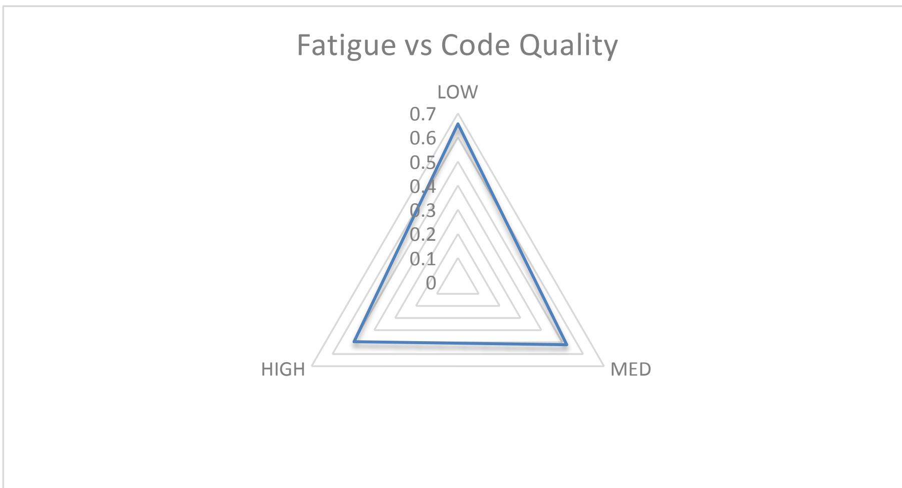


Figure 4.12 Radius graph comparing fatigue and code quality: more the fatigue level, less is the code quality.

In addition, we considered the Performance Score from the DevFatigue daily survey. The question is a part of NASA-TLX and says, *"How successful do you think you were in accomplishing the goals of the current task? How satisfied were you with your performance in accomplishing these goals?"* Fritz et. al. showed in their study that setting goals and completing those goals is considered a measure of productivity by software developers [Mey14]. As the question speaks about the **outcome and goal** of the project, we considered it as a measure of Productivity. We expected that there will be a drop in the outcome with the increase in fatigue level, which can be observed in the Fig. 4.13, hence supporting our claim with a Spearman's correlation coefficient of -0.9655674. This test examines the relation of fatigue and productivity. We, therefore conclude that a fatigue mental state affects the code quality and outcome of a software. Moreover, features like code analysis and the goal achievement can be used to measure productivity.

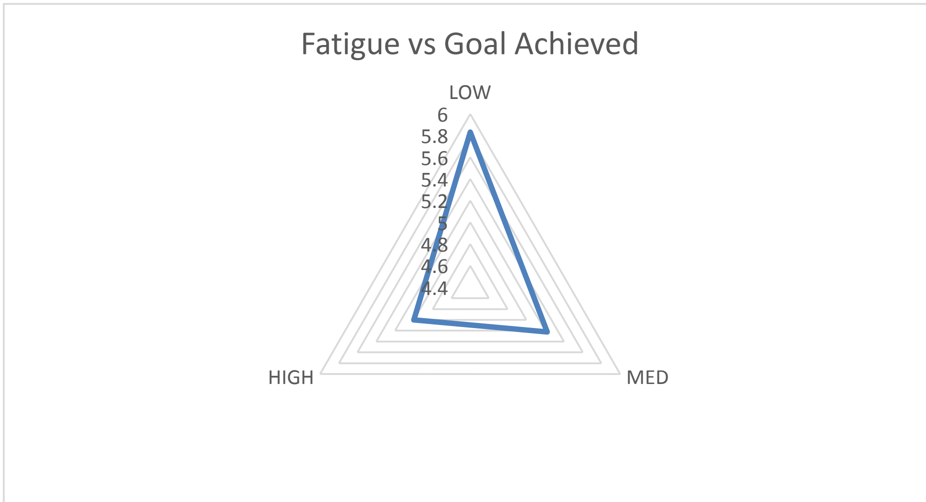


Figure 4.13 Radius graph comparing fatigue and goal achieved as reported by the participants in the Dev-Fatigue daily survey: more the fatigue level, less is the goal completion.

4.3.4 Errors/Mistakes

Since previous literature [Pim13] [Pim14] suggests an increase in the number of errors per key and loss of performance in case of fatigue, when working on computing general tasks. We expect similar for software development activities and claim that there will be an increase in errors and corrections per key.

To answer RQ6, we examined the **errors and corrections per key** in terms of usage of delete and backspace keys. Apart from the corrections made using keys, we have considered the corrections using the commands provided by the IDEs. Figure 4.14 shows the relation between fatigue levels and the error features. The graph supports our claim as it clearly shows an increase in error rate with an increase in fatigue level. We calculated the Spearman's Correlation Coefficient to be 0.732263 and 0.697866 for key based and command based error rate respectively. We calculated the key error rate by finding out the ratio of errors with respect to the total key usage. Similarly, the command based error rate is calculated with respect to the total commands used. As observed, in key error rate, the value of high fatigue is less than that of the medium fatigue level. This is due to the fact that there are some cases of no errors/corrections and nil or negligible key usage in high fatigue level, because developers tend to work less in higher fatigue level. The divergence in the empirical result can be avoided with large data set. Based on the statistical analysis result, we can conclude that our hypothesis stands correct. We may observe a stronger correlation in the future with better modeling

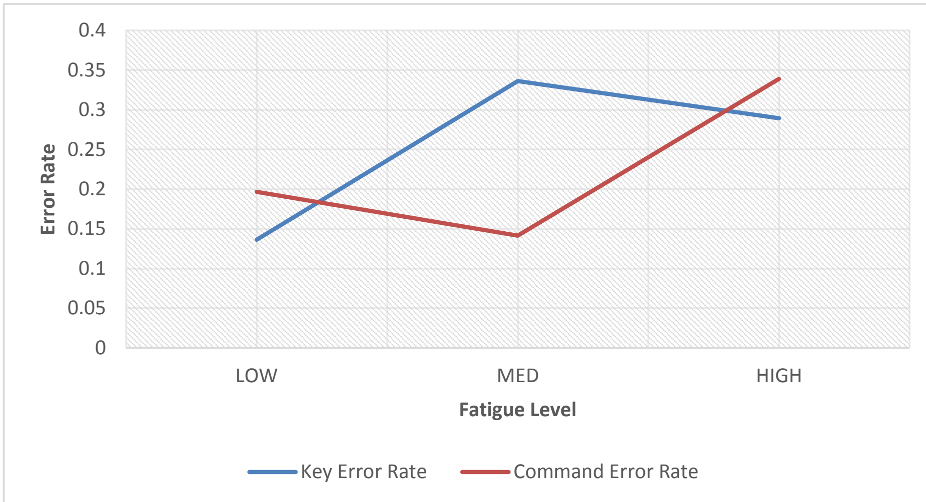


Figure 4.14 Line chart comparing key error rate and command error rate to fatigue level: more the fatigue level, more the error rate.

of the effect of low activity tasks during fatigue sessions.

4.3.5 Mental Work-set

Mental Work-set is defined as working on multiple tasks simultaneously. It is developer's capacity to work on concurrent activities at a time. We expect developer to work on less number of files in a fatigued state. We found out that the number of files are more in higher fatigue levels, hence disproving our hypothesis. If we analysis the results, more number of files traversal makes sense as a developer would be more browsing and less coding. Hence it answers RQ7 and conclude that there will be more number of files in case of higher fatigue level.

As our hypothesis did not hold true, instead of considering number of files, we examined the **files per time rate**, which would calculate the numbers of files visited in a particular time, shown in Fig. 4.15. In addition, we claimed that there would be an increase in systematic manipulation with increase in fatigue level. 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 code editing. A fatigued developer would depend more on the systematic manipulation. Figure 4.15 also shows that there is an increase in **systematic manipulation** with increase in fatigue level supporting our hypothesis. We performed a statistical analysis using Spearman's correlation coefficient on the aggregated values of both the features files per time and systematic manipulation with respect to fatigue level and the found the results as 0.943656 and 0.995512, respectively, which support our claims. Furthermore, for the Files per Time set of data, we calculated the t-test score and found that the data sets for each fatigue level is different from each other with a t-test score of: Low vs High = 0.104542, Low vs Med = 0.239363, and Med vs High = 0.905439.

This result answers the question in RQ8 and therefore concludes that fatigue affects mental work-set in terms of features like file traversal and systematic manipulation.

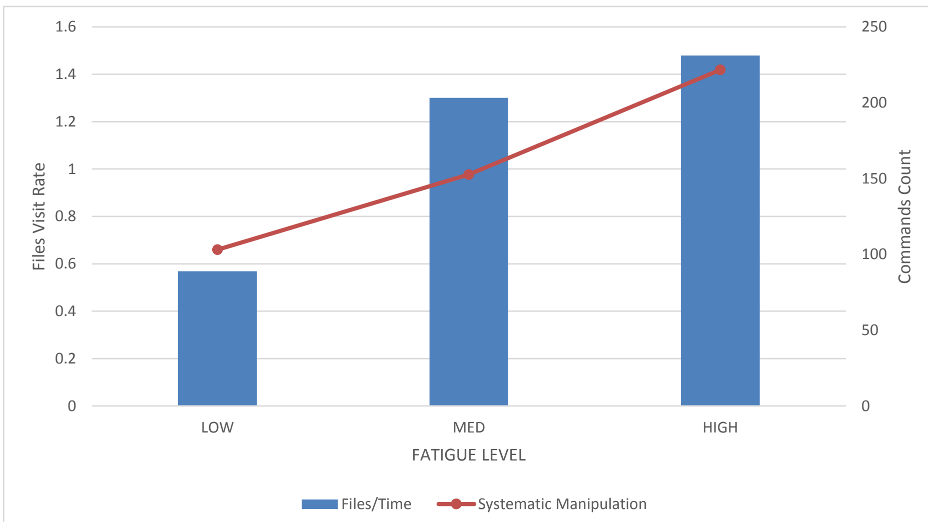


Figure 4.15 Bar plot comparing between Files per Time with fatigue level: in a high fatigue level day, a developer would browse more files. A line chart comparing systematic manipulation with fatigue level: more fatigue level, developer would depend more on the IDE provided tools than real coding.

4.3.6 Motivation

Figure 4.16 shows a relation of fatigue levels with **user activities, key usage, mouse usage and time spent** on software development taking an average for each day. In the graph, the primary Y axis is the activity count and secondary Y axis is the time in minutes. The X axis denotes the fatigue levels. The data is an aggregation of all the 9 users over the period of 7 days. We observed that our hypothesis that key usage, mouse usage, total activities and time spent would decrease with increase in fatigue level, is supported by the graph. We calculated the Spearman's correlation coefficient and the results are −0.99188, −0.97745, −0.87979 and −0.99993 for time, user activity, key usage and mouse usage, respectively. Additionally, for the User Activities data set, we calculated the t-test score and found that the data sets for each fatigue level is different from each other with a t-test score of: Low vs High = 0.409724, Low vs Med = 0.450093, and Med vs High = 0.450093. We observed that the difference is not significant as then it would have expected a t-test score value of less that 0.05 but the calculated score here shows promising results with more amount of data. The graph also helps us in addressing the question in RQ9, as we examined the graph and found out that the ratio of key usage to mouse usage decreases with increase in fatigue level. Hence, comparatively the decrease rate of key usage is less than that of the mouse usage. We determined that this analysis is useful in answering RQ3 as well. Moreover, we found out that fatigue also affects the time spent on software development tasks, which answers RQ10. We therefore conclude that fatigue affects motivation which makes developers work less. We also found out that the features like time spent and user activities can be useful in detecting fatigue. We could state that in a software industry working less would also be considered as low productivity, similar to low code quality.

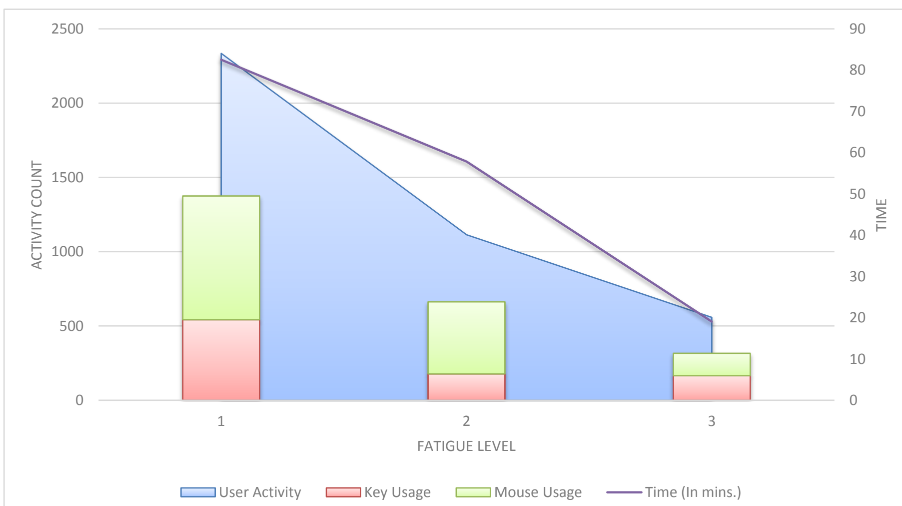


Figure 4.16 Stacked bar plot comparing key usage with mouse usage with respect to fatigue levels: in high fatigue level, developer would tend to work less and mouse usage would be more than key usage comparatively. A line chart showing the relation between time spent and fatigue: as expected, more fatigue would make a developer work less.

Apart from analyzing the effects of fatigue, we analyzed the relation between fatigue and other aspects of a task (NASA-TLX Score). TLX score could be used to evaluate a task difficulty. Figure

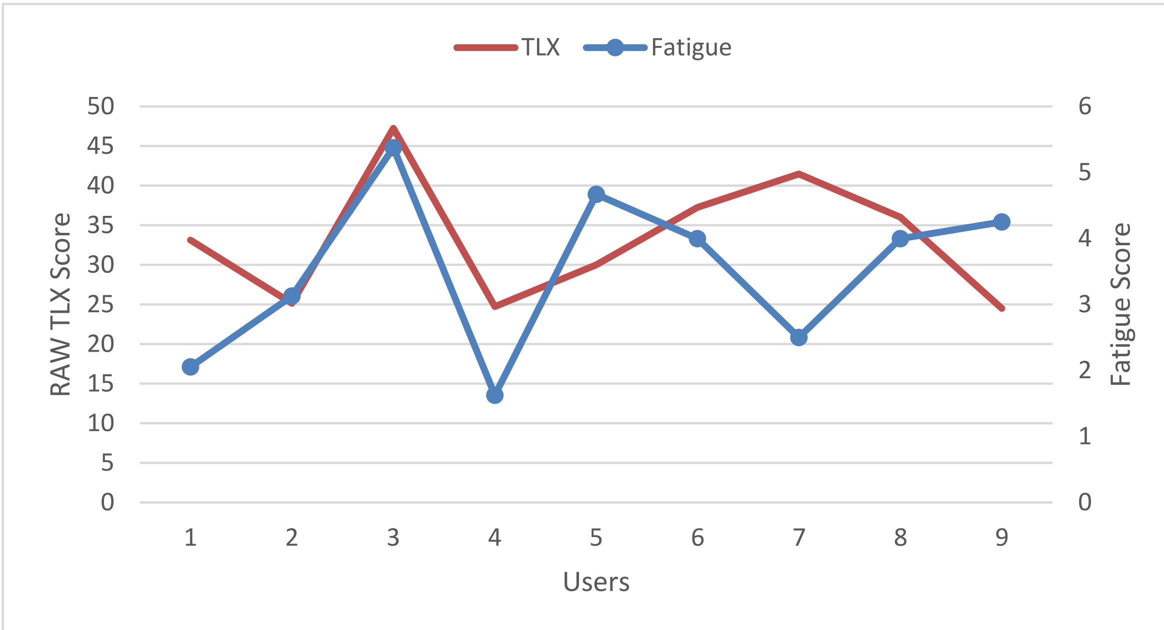


Figure 4.17 Line chart comparing the fatigue score to the overall TLX score for all the participants: TLX score would diverge with fatigue some times proving that the other factors also matter sometimes.

4.17 depicts a comparison between the task difficulty and fatigue as a separate factor for all the 9 participants for the whole study period. We observed that for some users relation between the fatigue and TLX score is diverging. For example, there are times when a developer would only take on a "hard" task if he is feeling "awake". We found out that in some scenarios task difficulty can be a distinct factor from fatigue.

4.4 Threats to Validity

Similar to the survey study, there are some threats to validity in our observation study as well. The small number of participants in our observation, the use of personal contacts for inviting some of the participants and the period of the study capturing a total of 7 days of work might limit the generality of the results of this study. We tried to address this threat by selecting participants from different companies and demographic location to cover a diverse population. Additionally, the participants were observed during their normal real-world work and not during an controlled experimental setup. Another limitation might be the study period of all the participants. Some of the participants could not complete the 7 days of study due to some technical, official or personal issues. That includes missing data in terms of attributes as well for some days. For example, on a high fatigue level day, the user activities are too less to extract data out of it. Even there were days when the user didn't have any activities despite of reporting low fatigue level. We, therefore, asked the other participants to extend their study period by a day or two. Moreover, we used the approach of categorizing the fatigue in different levels according to their scores and used the aggregated average for analysis.

The collection and categorization of the data poses another threat to validity, since fatigue could have been perceived differently by all the participants. When asked to score their daily work, every one could have had a different criteria in their mind. A participant X's fatigue score of 3 could have been similar to another participant Y's score of 5. This applies to all the TLX attributes. We tried to mitigate this risk by considering every one on a same level. Furthermore, as reported, some of them had sleep disorders. As we didn't have lot of participants, a categorization on the basis of sleep disorder would not have added any value. So, we didn't consider this attribute in analyzing the data and treated all on a comparable measure. Additionally, people are highly motivated in short-term studies and it may be difficult to measure and project the difference in actual performance.

Finally, in case of a controlled in lab study, the tagging of the data could have been more precise using physiological devices. In our study, we relied on the developer's recognition of assessing their work.

CHAPTER

5 DISCUSSION

Our findings contribute new knowledge about fatigue and how it relates to software development.

5.1 Survey

Apart from the discovered factors, studies [TN04] have shown that environmental factors such as peer behavior, experience, knowledge of programming languages etc. also affects productivity of programming tasks.

Around 50% of our participants were between the age of 20 to 35, implying that they are aware of the current trend of the industry and of the work culture followed in software companies. Rest of the respondents were older implying experience. As an online survey, respondents might be people from different roles of software industry with different and sometimes conflicting opinions. Some respondents [5.63%] disagreed to the statement that tiredness influences their work at all. Responses say that they are habituated to long working hours, which does not affect their efficiency. Less sleep is common among some developers and they still tend to be productive. Nonetheless, working for long productive hours can be normal, but it leaves residues/impact on other following tasks. Some respondents opposed the statement of fatigue affecting programming, saying *"all of a sudden an idea strikes and I am rejuvenated and fresh. Where as there are other times that even if I* *am fresh I might feel dull. So I feel that tiredness does not effect my performance.*" The reasoning was indirectly pointing out **Motivation** which is one of the already defined factors. There may be other cognitive mechanisms that may help developers overcome the sensation of fatigue, which are worth exploring in future work.

Although, each of the factors are intertwined in a way, it can be seen in the responses that respondents think that mental fatigue affects **Performance** the most. More than 100 responses included multiple factors. For example, when user is not motivated they tend to work less, they get distracted by other trivial tasks not able to work on a large work-set leading to a low performance and productivity. Some respondents stated, *"Take longer to complete a task. Hold less information in my head. Easily distracted. Make more silly mistakes."* Considering this as one of the research guidelines, we conducted an observational study monitoring the work pattern of some developers over a period of time.

5.2 Observation

Our findings contribute new knowledge about how fatigue is related to a deprivation in developer's efficacy and affects software development. Despite the concern of the industry about the quality of software products, effects of fatigue has not yet well explored. The results achieved from the observation study not only answer the research questions derived from the effects of fatigue, but also, provide evidence that it is indeed possible to analyze and quantify developer's fatigue level through features collected from monitoring their activities. We observed there was a trend towards decrease in performance and productivity with the onset of fatigue and figured out the key predictors which represent the factors affected by fatigue and can be utilized to detect a developer's fatigue state.

These predictors supports the fundamental understanding of fatigue related to software related activities. Our approach relies on the behavioral analysis rather on traditional physiological sensors, being non-invasive and non-intrusive. All in all, such approaches can have a positive effect on both the developer and the company he is working for. Nonetheless, we acknowledge that more accurate approaches can be developed when considering specific context information not yet considered, such as the type of task, factors like caffeine consumption, effect of sleep or food or other environmental factors discussed earlier. These approaches can help the future researchers to come up with a prediction model for detecting fatigue according to user's work pattern. Figure 5.1 shows a heat matrix which gives us a picture of all important predictors/features for building such a model, supported by our study. Though an approach of a controlled experiment using physiological sensors such as pulse detectors and eye tracking devices could provide a different platform for tagging and analyzing data.

Features

Factors	TLX Score	Key Usage	Mouse Usage	Session Timings	Commands	Files	Methods	Code Analysis
Performance		✓	✓	✓				
Focus		✓	✓	✓	✓	✓	✓	
Productivity	✓							✓
Errors		✓		✓	✓			
Mental Workset				✓	✓	✓		
Motivation		✓	✓	✓				
	✓	✓	✓	✓	✓	✓	✓	✓

Figure 5.1 Heat matrix showing the key predictors to build a model for detecting developer's fatigue in industry.

5.3 Interventions

In the daily life of a software developer, they frequently have the inclination of being depleted because of mental or physical work and a feeling of degradation in the quality of the simplest tasks. There are days when a developer's ability of performing a task is at its worst or some days when his performance diminishes slowly as the day advances. This deprivation in the efficacy, whether physical or mental, is linked to the onset of fatigue. Fatigue is evoked from repetitive demanding tasks. Nonetheless, 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 fatigue. The main goal of this study is to monitor and collect data in a non-invasive way and present the results in a cordial manner.

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[1] 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."*

Developer's fatigue often occurs when the programming team miss their deadline or because of

[1]http://pomodorotechnique.com/

tight schedules as a result of urgency. Now, they are tied to their desks servicing numerous tweaks, requests, and bugs with no formal expectation of when they will be out of the woods. A good timeline indicates a well-defined time or duration by the end of which the team expects to be done with a set of tasks and ready to move on to the next. It guides a team in deciding how to ration their energy, their workday, their private lives, and so on. Change orders, rush fees, and lift and shift timelines are part of administrative controls to align the broader team. The awareness of developer's fatigue level may play a crucial role in providing a decision-support system that will improve the quality of their life and thus work.

The next step for the future researchers and developers would be to use the predictors and develop a tool that could be used to monitor and report fatigue in industry. The main idea here is to build a classification model to detect fatigue, and that would need a training data. The tool can work in two phases: training and testing. In the training phase, we can have the user work on the tool for a period of time, letting the tool collect his activities and interaction performing the same analysis we did in our observation study. This analysis will help the tool to come up with a metrics for each of the defined factors. The tool can make a user profile including the predictors and their corresponding metrics for a particular user. The profile would include his working patterns in terms of relation between the predictor values and their corresponding fatigue level. For example: If the value of files per time is in some particular range for a day that triggers a high fatigue level for that user or a file/method transition pattern that could be classified in high fatigue level. These kind of analysis would report a fatigue state.

A framework of such a tool involves four steps: monitor, collect, detect and report. The idea is to have different processes of the framework performing different tasks. For example, the first layer would be data acquisition which would monitor user's activities and collect the required predictors shown in Fig. 5.1, like session timings, key interaction, mouse usage, files, code analysis results, etc. This layer would help encoding each activity with necessary information like time-stamps. The second layer would be data processing. In this layer, the data collected would be transformed to be processed. This layer would filter outliers and some data with negative affect on the analysis. The Classification layer would try mapping the evaluated metrics with already existing metrics and come up with a classification result. And the last Reporting layer would report the fatigued state. The analysis and reporting could be done in various ways. One would be analysis and classifying a whole day and flagging the code for further inspection in case it was a fatigued day. The other type of analysis would real-time, in which user would be analyzed in terms of each session or each activity and would be reported instantly suggesting some coping mechanism like freezing the screen for letting the user take a break and reduce the effect of mental fatigue.

This research holds a value in the industry setting some design guidelines and aims to help the developers get rid of their fatigue state.

5.4 Conclusion & Future Directions

The study of mental fatigue, including its causes and symptoms, is traditionally supported by data collected through instrumentation, self-reporting mechanisms (generally questionnaires) or, more recently, through the use of physiological sensors.

To comprehend the mental fatigue of software developers, we conducted a survey and card sort to catalog factors that are affected by fatigue. We then conducted a formative study to detect fatigue in software industry with respect to all the defined factors and came up with all the predictors that could be considered as some fundamental guidelines for building a model to detect fatigue automatically in industry.

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.

This study is first of the many steps taken in this direction, opening a wider scope for more research in the domain, such as, development of real-time systems for fatigue monitoring and classification through the defined predictors. Moreover, models could be built depending on user specific patterns to identify them. We hope this study will encourage similar approaches to continue investigating different ways to alleviate mental fatigue.

BIBLIOGRAPHY

[Aar99] Aaronson, L. S. et al. “Defining and measuring fatigue”. *Image: the journal of nursing scholarship* **31**.1 (1999), pp. 45–50.

[Bar09] Barker, L. M. “Measuring and modeling the effects of fatigue on performance: Specific application to the nursing profession”. PhD thesis. Virginia Polytechnic Institute and State University, 2009.

[Bau08] Baulk, S. D. et al. “Chasing the silver bullet: measuring driver fatigue using simple and complex tasks”. *Accident Analysis & Prevention* **40**.1 (2008), pp. 396–402.

[BZ14] Begel, A. & Zimmermann, T. “Analyze this! 145 questions for data scientists in software engineering”. *Proceedings of the 36th International Conference on Software Engineering*. ACM. 2014, pp. 12–23.

[Bur07] Burn, O. *Checkstyle*. 2007.

[CR07] Coetzer, G. H. & Richmond, L. “An empirical analysis of the relationship between adult attention deficit and efficacy for working in teams”. *Team Performance Management: An International Journal* **13**.1/2 (2007), pp. 5–20.

[Dan11] Dan. *Top 10 Symptoms of Developer Burnout*. http://tech.onthis.net/2011/06/16/top-10-symptoms-of-developer-burnout/. 2011.

[FT03] Folkard, S. & Tucker, P. “Shift work, safety and productivity”. *Occupational medicine* **53**.2 (2003), pp. 95–101.

[HS88] Hart, S. G. & Staveland, L. E. “Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research”. *Advances in psychology* **52** (1988), pp. 139–183.

[Kah70] Kahneman, D. “Remarks on attention control”. *Acta Psychologica* **33** (1970), pp. 118–131.

[Kha11] Khan, I. A. et al. “Do moods affect programmersâĂŹ debug performance?” *Cognition, Technology & Work* **13**.4 (2011), pp. 245–258.

[Lar97] Larson, G. E. et al. “Further evidence on dimensionality and correlates of the Cognitive Failures Questionnaire”. *British Journal of Psychology* **88**.1 (1997), pp. 29–38.

[Leh05] Lehman, A. *JMP for basic univariate and multivariate statistics: a step-by-step guide*. SAS Institute, 2005.

[Mak11] Makabee, H. *Effective Software Design*. http://effectivesoftwaredesign.com/2011/08/23/how-decision-fatigue-affects-the-efficacy-of-programmers/. 2011.

[Mey14] Meyer, A. N. et al. “Software DevelopersâĂŹ Perceptions of Productivity”. *SIGSOFT FSE, to appear, ACM* (2014).

[MM96] Morris, T. & Miller, J. C. “Electrooculographic and performance indices of fatigue during simulated flight”. *Biological psychology* **42**.3 (1996), pp. 343–360.

[Muc06] Muchinsky, P. M. *Psychology applied to work*. Cengage Learning, 2006.

[Nan12] Nanghaka, D. *Developer Fatigue*. http://dndannang.blogspot.com/2012/07/developer-fatigue.html. 2012.

[PG06] Parnin, C. & Gorg, C. “Building usage contexts during program comprehension”. *Program Comprehension, 2006. ICPC 2006. 14th IEEE International Conference on*. IEEE. 2006, pp. 13–22.

[PR12] Parnin, C. & Rugaber, S. “Programmer information needs after memory failure”. *Program Comprehension (ICPC), 2012 IEEE 20th International Conference on*. IEEE. 2012, pp. 123–132.

[Par14] Parsons, K. *Human thermal environments: the effects of hot, moderate, and cold environments on human health, comfort, and performance*. Crc Press, 2014.

[Pet09] Petrillo, F. et al. “What went wrong? A survey of problems in game development”. *Computers in Entertainment (CIE)* **7**.1 (2009), p. 13.

[Pim13] Pimenta, A. et al. “Monitoring mental fatigue through the analysis of keyboard and mouse interaction patterns”. *Hybrid Artificial Intelligent Systems*. Springer, 2013, pp. 222–231.

[Pim14] Pimenta, A. et al. “Analysis of Human Performance as a Measure of Mental Fatigue”. *Hybrid Artificial Intelligence Systems*. Springer, 2014, pp. 389–401.

[Sai99] Saito, K. “Measurement of fatigue in industries”. *Industrial health* **37**.2 (1999), pp. 134–142.

[SM79] Shneiderman, B. & Mayer, R. “Syntactic/semantic interactions in programmer behavior: A model and experimental results”. *International Journal of Computer & Information Sciences* **8**.3 (1979), pp. 219–238.

[SM86] Smith, A. P. & Miles, C. “The effects of lunch on cognitive vigilance tasks”. *Ergonomics* **29**.10 (1986), pp. 1251–1261.

[Smi93] Smith, A. P. et al. “Investigation of the effects of coffee on alertness and performance during the day and night”. *Neuropsychobiology* **27**.4 (1993), pp. 217–223.

[SK03] Smith, H. J. & Keil, M. “The reluctance to report bad news on troubled software projects: a theoretical model”. *Information Systems Journal* **13**.1 (2003), pp. 69–95.

[Sni14] Snipes, W. et al. “Experiences gamifying developer adoption of practices and tools”. *Companion Proceedings of the 36th International Conference on Software Engineering*. ACM. 2014, pp. 105–114.

[TN04] Tanabe, S & Nishihara, N. “Productivity and fatigue”. *Indoor Air* **14**.s7 (2004), pp. 126–133.

[Win05] Winwood, P. C. et al. “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* **47**.6 (2005), pp. 594–606.

APPENDICES

APPENDIX

A

SURVEY STUDY

A.1 Questionnaire

We are researchers [1] conducting a survey on fatigue and programming.

We previously studied other factors of in the workplace (e.g., Programmer, Interrupted [2]). Your answers will help researchers gather data about fatigue and investigate possible detection and alleviation mechanisms. This will also help us understand and investigate an opposite state of fatigue: **flow**.

1. Sleep Questions about your sleep habits and fatigue levels.
    - How many hours do you sleep typically, in a day?
    - How long did you sleep last night?
    - What are the factors, you think, lead to mental fatigue in your life mostly? (Check all that apply)
        - Stress

[1] http://www.chrisparnin.me/

[2] http://blog.ninlabs.com/2013/01/programmer-interrupted/

- Mood
- Sleepiness
- Physical work
- Non-willingness towards work

2. Fatigue at Work Questions about your fatigue and work performance.

   - Do you think fatigue is a severe and frequent problem for programmers?
   - When you are tired at work, you are most likely to do which of the following?
     (a) Continue working until the work is completed
     (b) Switch to other tasks until you regain your concentration
     (c) Stop working and rest
   - Do you feel when you are tired that it influences your work? If yes, what are some examples?
   - What makes to conclude that your performance is deteriorating or you need a break at the moment?

3. Work Questions about your work habits.

   - How long do you code in a day?
   - Describe your daily work routine. Does this routine occur during the morning, afternoon, or night?
   - What factors might reduce your energy/concentration level as you code, throughout the day?
   - Age

4. Other

   - Any other suggestions or comments?

APPENDIX

B

OBSERVATION STUDY

B.1 Pre-Study Questionnaire

B.1.1 Sleep Test

Complete this questionnaire to find out if you have sleep deprivation or other sleeping problems.

This test may help you recognize and detect symptoms of sleep deprivation related to sleeping disorders. The test is intended as a general source of educational information and does not contain medical advice. It should not be used for diagnosis or treatment. Getting an evaluation at a fully accredited sleep disorders center is the best way to determine if you have serious sleep deprivation related to a sleep disorder.

Q. When do you usually sleep? What is your daily sleep pattern?
Click here to enter text.

To take the sleep test, write down the number of each statement that is true for you. If the statement does not apply or is false, simply go on to the next statement. To score the test, follow the directions at the end of the questionnaire.

Group 1

1. I have been told that I snore.
2. I have been told that I hold my breath while I sleep.
3. I have high blood pressure.
4. My friends and family say that I'm often grumpy and irritable.
5. I wish I had more energy.
6. I get morning headaches.
7. I often wake up gasping for breath.
8. I am overweight.
9. I often feel sleepy and struggle to remain alert during the day.
10. I frequently wake with a dry mouth.

Total your score for 1-10:

Group 2

11. I have difficulty falling asleep.
12. Thoughts race through my mind and prevent me from getting to sleep.
13. I anticipate a problem with sleep several times a week.
14. I often wake up and have trouble going back to sleep.
15. I worry about things and have trouble relaxing.
16. I wake up earlier in the morning that I would like to.
17. I lie awake for half an hour or more before I fall asleep.
18. I often feel sad or depressed because I can't sleep.

Total your score for 11-18:

Group 3

19. I have trouble concentrating at work or school.
20. When I am angry or surprised, I feel like my muscles are going limp.
21. I have fallen asleep while driving.
22. I often feel like I am in a daze.
23. I have experienced vivid dreamlike scenes upon falling asleep or awakening.
24. I have fallen asleep in social settings such as movies or at a party.
25. I have vivid dreams soon after falling asleep or during naps.
26. I have "sleep attacks" during the day no matter how hard I try to stay awake.
27. I have episodes of feeling paralyzed during my sleep.

Total your score for 19-27:

Group 4

28. I wake up at night with an acid/sour taste in my mouth.
29. I wake up at night coughing or wheezing.
30. I have frequent sore throats.
31. I have heartburn at night.
32. During the night I suddenly wake up feeling like I am choking.

Total your score for 28-32:

Group 5

29. I have noticed (or others have commented) that parts of my body jerk during sleep.
30. I have been told that I kick and jerk during sleep.

31. When trying to go to sleep, I experience an aching or crawling sensation in my legs.

32. I experience leg pain or cramps at night.

33. Sometimes I can't keep my legs stiff at night, I just have to move them to feel comfortable.

34. Even though I slept during the night, I feel sleepy during the day.

Total your score for 33-38:

Sleep test developed by Russell Rosenberg, MD, Northside Hospital Sleep Disorders Center and Atlanta School of Sleep Medicine and Technology, 5780 Peachtree-Dunwoody Road, Suite 150, Atlanta, Georgia 30342-1611, 404/851-8135.

B.1.2 Scoring The Sleep Test

How to score your sleep.

Questions 1-10
If you answered YES to three or more questions, you have symptoms of **SLEEP APNEA** - a potentially serious disorder which causes you to stop breathing repeatedly, often hundreds of times in the night during your sleep.

Questions 11-18
If you answered YES to three or more questions, you have symptoms of **INSOMNIA** - a persistent inability to fall asleep or stay asleep.

Questions 19-27
If you answered YES to three or more questions, you have symptoms of **NARCOLEPSY** a lifelong disorder characterized by sleep attacks during the day.

Questions 28-32
If you answered YES to three or more questions, you have symptoms of **GASTROESOPHAGEAL REFLUX** - a disorder caused by acid "backing up" into the esophagus during sleep.

Questions 33-38
If you answered YES to three or more questions, you have symptoms of **PERIODIC LIMB MOVE-**

MENT DISORDER - uncontrollable leg or arm jerks during sleep or RESTLESS LEG SYNDROME-uncomfortable feelings in the legs at night.

B.2 Self-Assessment Survey

1. When did you last sleep and for how long?
   Click here to enter text.
2. Were you feeling enthusiastic before starting the task?
   Click here to enter text.
3. Was the task part of your homework or your own project?
   Click here to enter text.
4. Did you use any caffeine related products during the task like coffee, chocolate, tea etc.?
   Click here to enter text.
5. When did you last have any meal?
   Click here to enter text.
6. Type of task
   - Coding
   - Debugging
   - Planning
   - Design
   - Other
7. **Mental Demand.** How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
   Low: *0* ...... **High:** *10*
8. **Physical Demand.** How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
   Low: *0* ...... **High:** *10*

9. **Temporal Demand.** How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
   Low: *0* ...... **High:** *10*

10. **Performance.** How successful do you think you were in accomplishing the goals of the current task? How satisfied were you with your performance in accomplishing these goals?
    Low: *0* ...... **High:** *10*

11. **Effort.** How hard did you have to work (mentally and physically) to accomplish your level of performance?

12. **Frustration.** How insecure, discouraged, irritated, stressed and annoyed vs. secure, gratified, content, relaxed and complacent did you feel during the task?
    Low: *0* ...... **High:** *10*

13. **Fatigue.** How tired, exhausted, lethargic, sleepy vs. fresh, incapable and unproductive did you feel during the task? Do you plan to sleep now or continue doing some other stuffs?
    Low: *0* ...... **High:** *10*

14. **Motivation:.** How willing, motivated, interested and focused did you feel during the task? Did you feel you gained some knowledge by the task?
    Low: *0* ...... **High:** *10*

15. General Comments: (e.g. how your day was? What kind of work was assigned to you? How did you feel about it? Anything you want us to know!!)
    Click here to enter text.