EMOTION AND COGNITION ANALYSIS OF INTRO AND SENIOR CS STUDENTS IN SOFTWARE ENGINEERING

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Justin Evans

June 2021

COMMITTEE MEMBERSHIP

TITLE: Emotion and Cognition Analysis of Intro and Senior CS Students in Software Engineering

AUTHOR: Justin Evans

DATE SUBMITTED: June 2021

COMMITTEE CHAIR: Bruno da Silva, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Franz Kurfess, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Ayaan Kazerouni, Ph.D.
Professor of Computer Science

ABSTRACT

Emotion and Cognition Analysis of Intro and Senior CS Students in Software Engineering

Justin Evans

The software engineering community has advanced the field in the past few decades towards making the software development life cycle more efficient, robust, and streamlined. Advances such as better integrated development environments and agile workflows have made the process more efficient as well as more flexible. Despite these many achievements software engineers still spend a great deal of time writing, reading and reviewing code. These tasks require a lot of attention from the engineer with many different variables affecting the performance of the tasks. In recent years many researchers have come to investigate how emotion and the way we think about code affect our ability to write and understand another's code. In this work we look at how developers' emotions affect their ability to solve software engineering tasks such as code writing and review. We also investigate how and to what extent emotions differ with the software engineering experience of the subject. The methodologies we employed utilize the Emotiv Epoc+ to take readings of subjects' brain patterns while they perform code reviews as well as write basic code. We then examine how the electrical signals and patterns in the participants differ with experience in the field, as well as their efficiency and correctness in solving the software engineering tasks. We found in our study that senior students had much smaller distribution of emotions than novices with a few different emotion groups emerging. The novices, while able to be grouped, had a much wider dispersion of the emotion aspects recorded.

ACKNOWLEDGMENTS

Thanks to:

- My fiancee Haiyi for putting up with my shenanigans
- Dr. Bruno da Silva for being a great advisor and assisting me in being a better software engineer.
- Dr Franz Kurfess for letting me use the Emotive+ EEG headset and giving me a jumping off point to write this thesis
- Dr Ayaan Kazerouni, for helping me to become a better writer.
- My kids for being a nice distraction from all the work
- Lore Valenzuela for always having time to see if what I write made sense
- My parents for the weekly sanity checks
- My cat for keeping me company all the long nights spent in writers limbo

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The primary tasks required of software engineers is to write code and review others' code. In conducting these tasks, emotion, code understanding, and time can get in their way or alter the way that they can preform these tasks. One of the most important and often overlooked aspects that affects software engineers is the emotions they experienced while working [16]. As detailed in papers by Schwarz (2000) and Clore et al (2007), emotions change how we approach problem solving and affect our ability to make cognitive decisions [36][7]. The research in these papers comes to the conclusion that positive emotions are found to promote cognitive processing while negative emotions inhibit cognitive processing. While software engineers may not have the greatest say over when the need to perform tasks, knowing how their current state will affect their work will help them to understand the decisions they made at that time. There are also a lot of factors that can contribute to what emotions we feel. In terms of software development, familiarity with the code, code style, and experience may change the emotions that we have when working. This leads us to our first two research questions.

RQ1: Using a portable EEG can we find an association between the emotions the developers feel and their correctness and experience when performing software engineering tasks?

RQ2: What effect does code style have on developers' emotions and their level of comprehension while having to read and interpret code?

Humans express emotion in a wide array of forms with some being displayed visually such as a smile, while others can be more internal and long-lived such as regret. Emotion also comes in many different complex combinations with one emotion theory proposed by Silvan Tomkins detailing 9 basic aspects of emotion. These 9 aspects are: interest, enjoyment, surprise, fear, anger, distress, shame, contempt and disgust [34]. Modern sentiment analysis tools have started to incorporate these aspects to better break down the sentiment that they detect. While most sentiment analysis software is still relegated to outputting either positive, negative, or mixed emotion, active emotion detection using EEG, facial expression tracking and plethysmographs have expanded to incorporating some of these aspects.

Recognizing emotion in programmers as well as code sentiment has been a growing industry. In 2019 the market was worth around 21.6 billion USD with it expected to reach a market cap of 56 billion USD by 2024 [14]. This market is not just developers emotions but many other commercial applications. Emotion detected has been proposed for use in advertising, emergency response, and detecting burnout in the workplace. All of these applications utilize different technologies to achieve their goals but all look to see how the emotions of the individual would affect their behaviors and thoughts.

Being able to reliably detect emotion is not a simple task. Other works have been conducted on whether we can recognize emotions using electroencephalogram (EEG) software such as the research done by Daniela Girardi et al[17] [16]. These studies used the Epoc Insight and BrainLink headset which only have 5 channels and whose connection does not use saline sensors. Even with this limitation they were able to extract emotion data which lends credence to our studies ability to use an improved data stream via the Emotiv Epoc+ to read the emotions of developers. The Epoc+ has 14 channel saline sensors that are able to get better readings by measuring both the signals in the brain and the facial movements the individuals make. It also has ready built-in software that will convert the EEG signals into emotion data that can then be annotated and processed.

EEGs work by tracking and recording brain wave patterns. To do this they have small metal disks called electrodes that are placed on the scalp. The Epoc+ has 7 electrodes on each side of the headset that transmits the data wirelessly to a dongle connected to a computer. This allows participants to have more freedom of movement and comfort than many tests which utilize fMRI or wired EEGs. It is also much less expensive and can be used in any environment making it an ideal choice for testing on developer settings (sitting on regular desks and office chairs and in front of PCs or laptops).

The brain is broken into two hemispheres with each hemisphere having 4 lobes. Each of these lobes can then be further divided into their specific function with one lob (the frontal lobe) having regions for judgment, personality, speech, body movement, and self awareness [26]. These regions are not all used the same for each task and studies involving mice have shown that as the mice learn new tasks their brain activity changes over time as they advance from novice to expert[40]. Our study was conducted during the Covid-19 pandemic, and while this limited the distractions the participants had while testing it did bring its own challenges. While originally our study wanted to investigate the correlation of the regions of the brains that were activated with the experience of the developers and the tasks that they were doing. We found that we did not have the proper software to make this analysis. The BrainViz software did provide good visualizations but not enough of a readout to be used for classification.

By answering these questions we can analyze how programmers' brains respond to the process of software development by means of writing and reading coding. We expect that as someone is trained in software development their brain activity will change and the style and format of the code will have an impact on their emotion and ability to perform software engineering tasks. We aim at understanding how emotions programmers have while reading and writing code impact their comprehension. We also hope to see what effect different aspects of software development such as experience and code style has on the developer. These answers will help us to be more efficient with both the way we teach computer science and the way we perform when programming.

Chapter 2

BACKGROUND

There are many devices that researchers have used while analyzing developers for emotion and comprehension. The tools include devices such as fMRI's EEG's, eye -trackers and body sensors. There is also a growing use of tools to analyze sentiment embedded into code, and how the code style and lexicon changes with the developers overall sentiment. While I focused on just using an EEG for my research, oftentimes these devices are used together to provide better results [15].

2.1 The Human Brain

The human brain is responsible for all the complex thinking and decisions that we make. It works much like a computer sending and receiving messages to the body. It does this by utilizing both electrical and chemical transmission. Within the neurons, communication occurs using the movement of charged particles, while between neurons, in the synapses, the brain uses chemical transmission. These signals tell us what to do and feel.

The brain is divided into four lobes each with their own function, but some do overlap with others. The Frontal lobe, responsible for thinking, planning and problem solving. The Parietal lobe in charge of interpreting sensory information such as touch. The Occipital lobe, in charge of processing images and storing them in memory, and the temporal lobe, which processes information from your senses as well as dealing with memory storage [12].

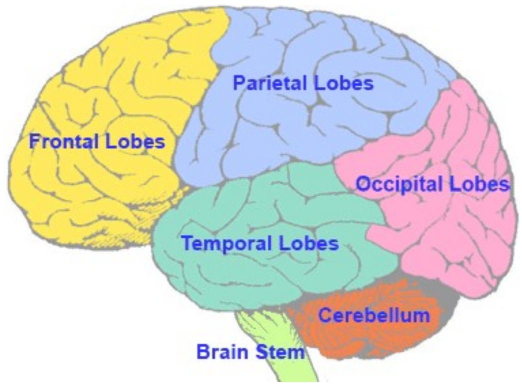

Figure 2.1: Human Brain with labels [32]

The mapping of the brain was completed in the early 1900s by German anatomist Korbinian Brodmann. He mapped the brain into 52 areas we call the Brodmann areas. These areas are still in use today with many key areas corresponding to a particular function. For example, Brodmann areas 44 and 45 are called Broca's area and are used for motor speech programming [11]. While the areas of the brain all correspond to a particular function, not every person uses these regions in the same way. When examining field hockey players, researchers noticed that experts in field hockey had greater activation of Brodmann areas 5, 17, and 18 when watching badminton videos [44]. The Brodmann areas provide a great reference for researchers trying to compare the brain activity between an individual as well as the brain activation patterns.

2.2 Electroencephalograms

The Electroencephalogram was first invented in 1875 by Richard Caton when he used it to measure neurophysiological recordings in animals. He was also the first to discover the presence of an electrical current in the brain. It was not used to study humans until 1924 when a German Psychiatrist Hans Berger developed a technique to use it to measure electrical activity in the brain[4]. The use of EEG's in computer science goes back over 60 years where some of its first uses was to measure alertness in individuals[21]. This and excitement have been found to be the easiest aspects to measure. It has since grown in use and can be utilized as a tool for controlling prosthetic limbs and interfacing with computers[42]. EEGs have been a mainstay in the neuroscience field and work by measuring the brain's electrical activity with electrodes. This electrical activity is made by the billions of nerve cells in the brain, with each producing a very small electrical signal. The EEG detects and amplifies these signals to be analyzed by a technician or software in our case.

The EEG's Electrodes have evolved through the years making readings both easier and more accurate. The first electrodes were concentric needle types, these were developed by Franklin Offner and were in use until after world war 2 [1]. It was in the 1950's that EEG topography was developed by William Walter that allowed EEGs to map electrical signals across the surface of the brain. Since the 1980s other more advanced technologies such as Blind Source Separation and Independent Component Analysis have been developed to improve EEG recording and electrode placement. Current electrodes do not need to be affixed to the scalp and instead use saline soaked felt tips to transmit the electrical signals to the electrodes and the EEG. This allows much easier testing and less consequences for the tester. The saline solution allows for the current to be picked up better allowing for a more useful brain image.

Figure 2.2: Emotiv Epoc+ [9]

EEG brain pattern maps look at the brain's electrical activity. They use the reading from the sensors as well as the signal frequencies received can make accurate mappings of what regions of the brain are used[8]. This has been achievable for decades now but with the advent of better electrodes and EEG devices is less intrusive allowing for more participation and research. The use of EEG's as the primary method to image the brain is rising noting the cost effectiveness and speed in which EEG's can be utilized in research[27]. The placement of electrodes on our device follows the 10-20 system which is aided by the software. Previous research criticized this system for its ambiguity but with the aid of the Emotiv software to ensure correct placement of electrodes we can ensure a quality signal [23]. We have create a picture of the placement as you can see in figure 2.3. We have color coded to match the lobes shown in figure 2.1.

The EEG that we plan to use, the Emotiv Epoc+, uses saline-soaked electrodes that are both more accurate and less invasive than previous electrodes that had to be glued on [9]. It has a total of 14 sensors which placement can be seen in the figure below. Other groups in software engineering research have used other emotive devices that use dry electrodes as well [16] [17]. These are not able to conduct and pick up the signal as well as the Emotiv Epoc+ nor do they offer the amount of contact points. By using the more advanced device we hoped to get better results from the study. The trade-off of using this device is that the participants would have wet electrodes pressed against their heads, but we did not encounter any discomfort amongst participants.

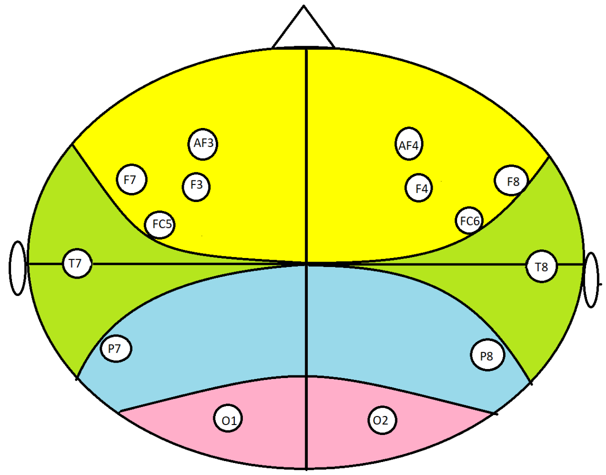

Figure 2.3: Emotiv EEG placements using 10-20 system

The Emotiv software will also break down the different signals that it receives and give us a read out of the emotion aspects detected. These range from 0 being low to 100 being high. The emotion aspects that it records are engagement excitement, focus, interest, relaxation and stress. These emotions are not exclusive with participants theoretically able to get 0 or 100 in all emotions at the same time[9].

The Emotiv device and software will allow for full brain visualization as well as a raw recording of the EEG data. The Emotiv BrainViz software breaks down the different frequency components read from the electrodes and displays where these signals came from on a 3D spatial map. The frequencies are color-coded on the diagram allowing one to know both where the activity is and what kind of activity it is [9]. Many studies have used technologies like this in the past but regularly pair it with other body sensors or eye tracking.

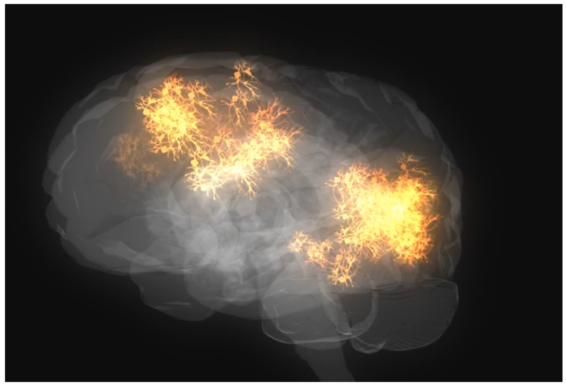

Figure 2.4: Emotiv BrainViz

2.3 fMRI

An fMRI device provides better resolution into the brain than EEG but requires users to be scanned in a large fMRI machine. This machine is a large tube with magnets that can pick up the magnetic signals in the brain utilizing nuclear magnetic resonance. This can be aided with the addition of a dye to improve contrast allowing for a better resolution of the image. The images recorded show the difference in blood flow to regions of the brain which can then be correlated to which regions of the brain are activated[33]. These readings are highly accurate, but the devices are both expensive and limiting as research tools. Since it uses high powered magnets anyone with metallic prosthesis or implants cannot take part in this type of research. The technique most used, blood-oxygen-level dependent contrast was developed in 1990 and has come to dominate brain mapping research since no dye injections are required[43]. When using them for software engineering it limits the testing to reading code as one cannot produce code while in the fMRI machine.

Figure 2.5: fMRI machine [33]

fMRI research investigates many different neurological problems such as how the brains of people with schizophrenia differ from regular people [20] as well as research close to that done here. Much of the research conducted with fMRI's look at what regions of the brain are activating and correlating that to what the participant reported thinking about or was doing. Such as in the case of West et al's. work. In this research they identified regions of the brain that perceive different emotions such as sadness happiness or anger [24]. It is a powerful tool that can do many things but does have a different use and its own restrictions.

2.4 Sentiment Analysis

The emotions and sentiments that developers feel while producing code has been investigated using many different mediums. Post sentiment analysis such as code sentiment analysis and lexicon analysis investigate what the developers might have been feeling by analyzing the sentiment of the comments and pull requests they commit when producing code. Sentiment analysis tools such as SentiStregnth-SE can be used to classify discussions on Stack Overflow or Jira as either positive, negative or neutral/mixed [46].

Sentiment analysis has a lot of commercial application. It can be used by companies to gauge customer sentiment over a product, detect bullying or hostile language on social platforms or help us to write emails in the correct tone [2]. This can then be used to make better products, protect people or ensure you are better understood. In research context a lot of research with these tools look into the accuracy of the sentiment and how to improve it in a viable way[31].

2.5 Other Technologies

Some of the most common tools used in conjunction with EEG sensors are the Empatica wristband and eye tracking software such as Tobii[13] [16][17] [25]. These tools when used with the EEGs provide more points of reference for the researchers to gather information and in some cases provide better accuracy than just the EEG headband alone[17]. The Empatica wristband contains a plethysmograph and sensors that record electro-dermal activity. By combining the movement of electrical signals and blood pressure with the patterns from a low resolution EEG one can fairly accurately predict the emotion of an individual[17].

In a recent survey paper researchers found that biometric measurements reflect cognitive load. This study also found many other studies that utilized eye tracking, galvanic skin response and heart rate monitoring. Many times these devices are used together, with a fMRI or EEG[18]. The aim of some of these studies is to use the level of cognitive load to predict the quality of code the software engineer will produce. By knowing this it may help software engineers prevent bugs from happening before they are even written [29].

Chapter 3

RELATED WORK

There have been multiple studies that investigate how programmers and novices think when conducting code reviews, or when answering simple programming tasks. While most of the research has been done using fMRI (functional magnetic resonance imaging), some recent studies have been turning to EEG as a solution, to allow the subject to perform the tests in their natural environment.The use of the EEG (electroencephalogram) allows users to move around and produce code rather than just review or answer simple questions.

3.1 Practical applications

Researchers have investigated the use of mobile EEG's to control electronics, while others have been using them to better understand the user. When looking at the research conducted into wearable technology and the application of EEG's for consumers there is still a spectrum of how these are incorporated. One paper by Jung et al. investigates the use of EEG's made into wigs for use in gaming [14]. They propose using the EEG to extract emotion data of the gamer for use in game. Separately the use of EEG's is being investigated for use in controlling user interfaces and making cybernetic wearables[3][10]. The use of EEGs in more practical applications is on the rise as research into brain to brain interfaces increases and the field of human computer interaction (HCI) expands. By utilizing EEG's researchers hope to make a Brain Machine Interface (BMI) that can be used by anyone. This would allow computers to be used as tools to grant accessibility and upward mobility for disabled users. BMI's further look into applications such as driving a car and operating machinery[10]. For these interfaces to work researchers have been training the EEG data on deep neural networks to build an accurate EEG decoding model. These models must be able to be generalized to every user and the models trained so far have shown to be competitive against supervised models specific for individuals[10]. EEG's build into wearables is a growing trend which we expect to see much more research conducted on in the future.

3.2 Emotion Research

While EEG's in wearables is a very practical approach the use of EEG's, to understand the users is also of vital research importance. EEGs have been used to understand a variety of human conditions. There has been research into human emotion, empathy, cognition and development[6][35][41]. Pratte et al. in their study of empathy measurement tools recommended incorporating EEG into the self reported surveys to get better understanding of the users empathy and feelings [6]. Research into cognition has given us insight into how we perceive color and recognising a driver's attention level while driving. This research into color perception has resulted in different ways that brain signals can be incorporated into wearables and IoT devices. The research into driver attention found we can measure the level of distraction in participants up to a 97% accuracy [35].

This cognitive distraction research can be further applied to measuring engagement in students and overload. This research was conducted by Nakagawa et al, studying student concentration while learning at home during the pandemic. This research did not have enough participants to be conclusive but did find that further study is warranted[30].

Research into the emotions of developers is ongoing but one of the most recent papers researching the emotions of developers was the paper by Girardi et al.[17] which used the empatica wristband in concert with the BrainLink headset. This paper sought to investigate the link between emotions and progress that developers make in their code. Finding that there was a link between emotion and the amount of progress that developers reported during the study. Other studies have been performed which gauge how emotions affect flow[28] to how well we can sense emotions[13][17].

3.3 Program Comprehension

Programmers utilize a lot of effort when trying to complete software engineering tasks. This effort is mainly in the form of cognition or mental tasks. It has been found that if software engineers use meta-cognition to deliberately choose a method to comprehend a piece of code they are able to perform better[37]. In order for software engineers to do this they must first better understand themselves.

There have been many studies performed by Dr. Janet Siegmund and her colleagues on program comprehension. Their research has shown great promise in identifying the regions of the brain that are used in program comprehension [38]. They were also able in a separate experiment to isolate specific cognitive processes that allowed for bottom up program comprehension. This research also found that beacons or program layout had no effect on the comprehension process[39].

The study of software developers brains and computer science students has yielded some good results in the past. By understanding how software engineers think we can better establish a pedagogical system to teach incoming computer science students. The research by Vierira and Farias suggest that by incorporating EEG data and biometric readings while students work on IDE's we could better design IDEs and teaching environments for developers[41]. This along with the emotions of developers has opened up new branches of neuroeducation which allow developers to better learn and work.

Chapter 4

STUDY DESIGN

Due to the Covid-19 pandemic the study had to be altered from its initial scope and premise to account for the lack of in person contact and availability of participants. This study was reviewed and approved by the CalPoly IRB board and testing only commenced once approval was given. Initially we had a pilot study of a senior level student and a novice level student to test our procedures. The results from these participants where not included in our data-set. We had hoped to preform testing in a large range of individuals to better analyze the change in emotion and cognition as students studied software engineering. While the study still was able to show the change, we took our sampling at the two end points of the students academic careers.

4.1 Participants

Participants were pulled from the student body at Cal Poly. Our beginning students were recruited from introductory computer science classes and clubs. We chose this as our novice vs seniors category since it is what we have most readily available. These students where given a questionnaire to gauge the level of experience they had with computer science. This questionnaire included questions such as "how many years of programming experience do you have?" and how they would rate their programming knowledge on a scale from 1 to 10. Students who had excessive experience (greater than one year) where not used in the study. The students typically came from Electrical Engineering, Computer Engineering, Computer Science and

Software Engineering majors but Computer science minors where also used for the intro computer science data-set.

For the senior level participants we recruited from the Computer Science Masters program and senior level courses in both Computer Science and Software Engineering. We also recruited faculty members who were willing to participate but this made up a very small portion of our data-set. For these participants we also looked at their performance on the software engineering tasks to ensure that they were all at an acceptable level of completion and accuracy. If the participant did not perform the tasks to an acceptable level of quality their data was not used in the study. This was chosen to be done as we wanted to compare individuals with a high level of knowledge with those of a basic level and thus only wanted high caliber senior level participants. In total we were able to recruit 11 Senior level participants and 10 novice level participants. One of the senior level participants data had to be excluded from the study based on the answers observed from the test.

4.2 Tasks

All the participants were asked to do the same tasks. These tasks where written up in such a way that a freshman at CalPoly could complete them but still make the senior level participants think about them. The first tasks the participants were instructed to complete was to solve the following programming question:

The function will be given an unordered array of integers named nums and a 2nd input of type integer called target. Return the indices of two numbers such that these two numbers add up to the target. Each of the output indices must be distinct. You may return the two indices in any order.

The participants were instructed that they could write the answer in any language of their choice or in pseudocode. The object of this task is to measure the emotion and cognition of the participants while they were producing code, in a language they were familiar with, as well as a gauge of their ability to perform software engineering tasks.

The other task that we had the participants complete was a code review of 4 different pieces of code. All of these snippets of code were written in Python as this is the first language taught at Cal Poly and thus would be accessible to all the students. These code snippets vary in complexity with Google's linter used to check style [19]. After all style errors where fixed 2 of the review samples where embedded with style errors such as bad naming conventions, magic numbers, long lines and bad spacing. The preceding figures in this section are the code snippets that we had the participants review. They reviewed the two samples without errors first then moved on to the samples with errors.

```
def code_review1(words):

    long_word_length = 5

    n = 0

    longest_word = ""

    for word in words:

        if len(word) > long_word_length:

            n = n+1

        if len(word) > len(longest_word):

            longest_word = word

    print(n)
```

Figure 4.1: Error Free Simple Code

```python
# CodeReview2 checks if queen is under attack on board

BOARD_SIZE = 8

def under_attack(col, queens):

    left = right = col


    for r, c in reversed(queens):

        left, right = left - 1, right + 1

        if c in (left, col, right):

            return True

    return False


# part of Code Review 2 recursively gets solutions so queens are not under attack

def solve(n):

    if n == 0:

        return [[]]

    smaller_solutions = solve(n - 1)

    return [solution + [(n, i + 1)]

            for i in range(BOARD_SIZE)

            for solution in smaller_solutions

            if not under_attack(i + 1, solution)]


for answer in solve(BOARD_SIZE):

    print(answer)
```

Figure 4.2: Error Free Complex Code

The figure of the code without errors deals with the 8 queens problem, which is a popular challenge question in many math classes at CalPoly. This code has minimal comments so that the review would have to study the code more in-depth. Though the code is without much supporting comments, the naming of the variables and functions is such that participants could figure out what the code is trying to accomplish.

```
"""code review
3"""
import sys,os,pandas,json
def codereview3():
   json_File = open('train.json', 'r'),
   f = open("ingredients.csv", "a"),
   values = json.load(jsonFile)
   i = 0
   while i < 10000:
     needCleaning = ( str ( values[ i ][ "ingredients" ])+ "\n" )
     needCleaning = needCleaning.strip( "[" ).replace("'" , "").replace("]" ,
"").capitalize().strip()
     f.write(needCleaning)
     needCleaning.startswith("Hap")
     i += 1
   jsonFile.close()
   f.close()
```

Figure 4.3: Simple Code with errors

The code of the figure 4.4 incorporates the following errors: Block comments not starting with just #, white-space after (, continuation line under-indent for visual indent, missing white-space after "," , multiple spaces before operator, new newline at end of file, bad inline comments, unnecessary spaces, direct comparison to True, bad naming convention, using O (oh) as single variable name. These errors where chosen with the addition of slightly more comments after pilot studies suggested that comments in code had a bigger impact on how reviewers rated code then the actual style.

```
#CodeReview4
def _Get_Closest_Centroid_(centroids,utterance,k):
    dist = 0#set distance to zero
    minDist = 1234567898#set min to big number
    closest = 0#closest is zero
    for O in range(0, k):
        dist = 0#zero out distance and check all k centroids
        for j in utterance:
            # calculate distance between Xj and all centroids
            if j not in centroids[ O ]:
                dist =( utterance [ j ] +
                dist)
        for word in centroids[ O ]:
            if (word in utterance) == True:
                dist =(dist -
                pow(centroids[ O ] [ word ] - utterance[ word ], 2))
            else:
                dist  =(dist +
                centroids[ O ][ word ] * 2)
        if dist < minDist:
            closest = O
            minDist = dist
    return closest
```

Figure 4.4: Complex Code With Style Errors

Participants were instructed while conducting these tasks that they could not go back to previous problems. The investigator monitored the participant through a one way mirror as well as following them along on a shared document accessible in the room the investigator was in. Participants were given guiding questions to help them with the code reviews. These questions are listed below and were used to gauge how well the participants understood the code. They also had the secondary function of getting them to think more deeply about the problems.

- How do you describe the code behavior?

- What would you change about the code?
- From 1 (bad) to 5 (excellent), how would you rate the quality of this code? And why?

4.3 Measurement Tools

In order to measure the participants emotion we used the Emotive Epoc+ EEG headset. This was the only tool used with the participants. This choice was made as we did not want to disturb participants with self reflections while they where conducting the tasks. In pilot studies not included in our data set participants corroborated the readings of the EEG while they were preforming the tasks. Before any tasks where started participants baselines where recorded and the device calibrated. This process usually took about 5 minutes but could take up to 15 for some participants. Participants were also asked to perform simple meditative exercises before-hand in order to get them into a more relaxed base state. These exercises included guided deep breathing and guided thought exercises. This helped to ensure that outside influences would have less of an effect on the study.

The Epoc+ is able to measure most of the active brain and utilizes signals that would indicate smiling, clenching teeth, smirking, laughing, blinking, winking, raising and lowering eyebrows and horizontal glances to classify the emotional levels of individuals [9].

4.4 Protocol

In order to eliminate as much bias as we could we have the same procedure for all participants in the study. After the participants are recruited and educated about their rights and nature of the study, they will sign a consent form before participating in the study. This form was in paper and explained what the testing was aiming to achieve as well as what they can expect by participating. It also informed them of their rights as a volunteer. They were then directed to the testing site located on Cal Poly's campus at an appointed time agreed upon by both investigator and participant. This testing site was chosen due to its convenience and the fact that it has a split room to aid with social distancing. Utilizing social distancing measures and proper cleaning protocols, we had users meet at the same location on the CalPoly campus. This split room has a window conjoining the two spaces allowing the investigator to easily watch participants while not disturbing them and maintaining social distancing. The participant stayed in the large room with a computer provided by the investigator in which to work on. At the start of the study the participant received assistance in donning the Emotiv Epoc+. If they had any questions about how the device works apart from what was already explained in the consent form they would be answered here. They were also informed that once the study starts it will either have to be abandoned or finished as breaks would disrupt the data collected. For the entire duration of the study they wore the Emotive device with the study consisting of 3 tasks. While performing these tasks the investigator stayed in the other room observing through the window only to ensure that if anything happens with the hardware the investigator could assist.

At the conclusion of the study, the participants exited the room and the data was annotated with all the activities that the participant did and at what time. The emotion data collected using the Emotiv app is saved on a drive to be converted. All data that could be tied back to the participant was anonymized with only notes indicating weather it was a novice or a senior. After this procedure the data is then processed and each of the participants data is separated into either novice or experienced categories. The emotions tied to each activity is then analyzed for different characteristics from the Emotiv app and stored in a CSV. The data can then be analyzed from the CSV format.

Chapter 5

DATA ANALYSIS

With the data collected I have an annotated data-set of the participants' emotions at different stages of their tasks. I then compared these data-sets using the data that I collect from the beginning survey to find a correlation between experience and the emotions developers feel when writing and reviewing code. Using the different code styles I also was able to gauge how code style impacts emotions and cognition in developers across knowledge levels. I then used a k-means clustering algorithm to cluster the participants into their respective cognition and emotional group to compare against recorded results. Finally, I then used this to tell if there is a correlation between the emotions felt when programming and one's experience in software engineering.

5.1 Data Collection

The data collected by the Emotiv Epoc+ headset was stored and accessed through the Emotive Pro app. For our testing purposes we were granted a Pro license through the month of June. This app allowed for the investigator to go back and review the recordings for both the EEG signal and EEG emotion data. The BrainViz recording was stored separately though in the Emotiv BrainViz software and screen recordings. All participants data was either marked novice or expert and a random id. The id could not be tied back to the individual but was used for comparing the notes taken by the investigator as well as the work they did in session.

The notes the investigator took detailed when the participant did different actions such as completing a task or if there was any unusual events that happened during testing. Due to Covid-19, campus was relatively empty, so there was little distraction during testing. If participants talked answers out loud or asked questions the investigator annotated, this also included events such as if the participant decided to get out of the chair, in the case of any sudden spikes in emotion or brain activity. These events where highlighted in the data recordings but no events that happened posed any significant shifts in the emotion data.

After all the data was recorded the investigator re-watched the recording and marked the 6 emotion levels that were recorded by the Emotiv device into excel. This was decided to be done at 20 second intervals after the investigator had previously done a round of data at 10 second intervals. The change between the ten second intervals and using a 20 second interval was less than a point. On the 100 point scale this was deemed an insignificant loss of accuracy of the data from the previous recording interval of 10 seconds. This data was grouped into two pages of excel with all the novices on one page and all the experts on the other. The investigator highlighted the cells in which the transition from one task to another took place. There was a total of 5 averages for each of the tasks completed for the individuals.

	69	70	78	62	65	79	66	65	57	68	60	54
	30	54	20	20	17	38	77	42	45	26	53	22
	70	75	60	53	49	58	60	59	57	43	57	42
	55	54	55	58	57	57	59	77	55	54	53	56
	33	29	34	37	38	29	37	28	36	32	36	35
xpert 857	81	93	85	85	81	78	68	60	62	34	62	61

Figure 5.1: Sample data-set with data changed task 1 and 2

57	65	41	50	53	51	44	46	57	50	42	56	68	50	66
20	58	27	30	10	20	27	23	44	33	10	6	20	57	7
39	44	44	36	36	30	43	41	27	39	45	22	25	26	34
63	70	68	74	71	53	53	69	76	58	59	54	54	58	51
32	30	29	34	26	27	42	37	34	33	47	32	33	40	28
43	49	34	29	34	19	35	42	19	38	62	17	29	26	30

Figure 5.2: Sample data-set with data changed task 3 and 4

56	56	56	40	53	47	65
38	23	38	11	8	14	17
29	34	31	29	34	21	28
49	53	61	52	55	49	67
31	28	34	33	34	36	40
21	20	27	33	37	19	18

Figure 5.3: Sample data-set with data changed task 5

Using BrainViz the investigator was not able to reliably extract the information and was forced to manually examine the recordings. The BrainViz software does not add any additional information, only providing a visualization of the signals already output in the Emotiv app. This was done by reviewing the recordings and looking at the different peaks in the EEG readouts from the Emotiv Pro app.

5.2 Data Analysis Protocol

The data gathered from the investigator was analyzed using multiple different methods. We averaged the emotion data results in excel and the standard deviation was calculated as well. This was done using the inbuilt excel formula totaling the cells and dividing by the total number of cells. These scores were calculated only for the different tasks the participant did and not over the whole testing period. Each participant had their own scores averaged over the tasks. There were also groups created with overall task averages so that the investigator could more accurately compare the expert and novice data-set.

We fed the averages of the participants into a python program that used the averages as weights.This program used a k-means algorithm that went through 100 iterations with 2-4 grouping clusters. Each iteration started with a random initialization of the centroids. This procedure was run through for each of the number of clusters and 10 different runs to verify group distribution. Using the k-means algorithm we where able to differentiate the seniors group from the novices when both where fed into the algorithm. Since we wanted to do a further breakdown we then separated the data into novice and seniors to find a further breakdown of the groups. When we broke down into novices and seniors we ran the clustering algorithm with a choice of 2 grouping clusters.

The choice to use 2 grouping clusters was chosen since we had 2 groups from which we collected our data. Utilizing the elbow method for determining the number of clusters we got 4 and thus was the primary measure we used when we analyzed this output. 3 was also done as the data set was not significantly large and we wished to see how the number of clusters affected the grouping of the data-set.

When looking at the novice groups, the emotion distribution is much more scattered. Nevertheless there was similarity enough to form two different groups when analyzing their data. The membership to these groups which we will call Novice 1 and Novice 2 is included in the table below.

Table 5.1: Group Membership

Novice group 1	Novice group 2	Senior group 1	Senior group 2
Novice 853	Novice 23	Senior 336	Senior 23
Novice 2	Novice 3	Senior 861	Senior 008
Novice 805	Novice 291	Senior 42	
Novice 976		Senior 22	
Novice 745		Senior 59	
Novice 966		Senior 8742	
Novice 777		Senior 194	
		Senior 614	

The senior groups are likewise clustered into two groups. The grouping of all of these members is much tighter than the novices so much so that it could be treated as one large group. The decision to keep the elbows recommendation and have two groups is there are some parts of the data-set in which these two groups diverge that were significant. The two groups will be called Senior 1 and Senior 2 with the membership of Senior 1 being 8 members and the membership of senior 2 being 2 members.

This was useful for verifying what we were seeing in the excel spreadsheet. Within the two groups their where particular emotion groupings that started to emerge with this algorithm further showing us which points were "close" to each other. These patterns of emotion existed in both the novice and the senior participants. We talk more about this in the results.

While the average and standard deviation were the primary measure we used to compare the data, the range and median of each of the tasks where also recorded. During all of the task most of the emotions experienced some spikes, which is why the average emotion of the task was chosen to be used as the primary measure. It also appeared to give the best basis of what emotions the participants were experiencing while doing each of the tasks.

5.3 Emotion Results

Using the Emotiv Pro software we where able to extract emotion data to be analyzed. The emotion data of the novices and the seniors showed significant variance when looking at the groups that emerged. Each of the Tasks affected these groups differently with Novices experiencing higher overall emotion during the testing.

5.3.1 Task 1

The first task that we had the participants do was to solve a simple programming question. This allowed them to produce their own code or psudocode and across all groups of participants we saw very similar engagement levels except for the second senior group. This group saw an extremely high level of engagement that would not be seen again by any other group in any of the tasks.

Table 5.2: Task 1 Group Results

	Novice 1	Novice 2	Senior 1	Senior 2
Engagement	78.32	69.42	68.57	91.80
Excitement	62.65	15.17	39.79	53.48
Focus	49.18	26.64	56.66	48.57
Interest	69.29	65.75	55.01	61.33
Relaxation	37.60	67.53	29.79	28.71
Stress	47.24	63.67	57.49	47.90

Excitement across the novices varied greatly with a standard deviation of 25.1 across Novice 1 and an extreme gap between Novice group 1 and Novice group 2. It should be noted that Novice group 2's smaller excitement score also comes with a much smaller standard deviation of only 8.15. Novices tended to be both more excited and less excited than their senior counterparts. Since the data was anonymized the reason for this can not be made other than mere conjecture. It should be noted though that the novices were pulled from a more diverse pool than the seniors. When looking at the excitement that the groups experience when producing their own code we can see that seniors tend to be moderately excited when preforming this task.

When looking at the focus attribute we can see that, except for Novice group 2, all the groups averaged moderate focus while preforming this task. The focus within the other three groups with the highest standard deviation coming from Novice group 1 with 7.36. This changes when looking at group 2 who has a standard deviation of 17.92 when reviewing the focus attribute. When looking at the results and accuracy of the programs written for this task those from Novice group 2 wrote significantly worse programs than any other group.

We see a similar trend with relaxation as we did with focus. Most of the seniors all having low relaxation scores and a mix when looking at the novice groups. Novice group 2 is again an outlier with moderately high relaxation when compared against the other groups. Many of the participants in Novice group 2 also took longer to complete the first section then when looking at the other groups with the longest time in novice group 2 being 16 minutes for the first task. The average time to complete this task was 4 minutes 43 seconds.

Interest across the board was very similar for all participants. With novices having higher interest levels than seniors but not as significantly as the other emotion aspects. Overall the seniors were less relaxed, excited and less interested than their novice counterparts. Seniors tended to focus more when writing their programs and all of them where able to come up with a viable solution to the question posed in task 1.

5.3.2 Task 2

Task two is the first of the code review questions. The code review set of tasks had two complex code snippets and two simple code snippets. For task 2, it was a simple code snippet with no style violations. We gave all participants the same code and questions to review. The novices ended up taking more time on this section averaging around 6 minutes while the senior participants averaged around 4 minutes.

Moving from code production to code review did change some of the seniors' emotion levels. Seniors across the board had lower stress levels when conducting the first code review task. This can be due to a variety of reasons such as the simplicity of the code meant to review or the pressure relived from not having to come up with their

Table 5.3: Task 2 Group Results

	Novice 1	Novice 2	Senior 1	Senior 2
Engagement	68.88	75.42	66.34	72.46
Excitement	46.91	17.57	51.18	54.20
Focus	52.04	13.14	57.33	53.40
Interest	62.44	61.30	57.69	61.93
Relaxation	32.43	71.43	29.38	28.06
Stress	52.62	42.28	48.59	38.86

own solutions. For most novices this will most likely be their first time conducting a code review (on someone else's code) which could be the cause for the increase in stress from Novice group 1. The Novice group 2, which did the worst in the code production, also did the worst in this task. The novices in this group show a decrease in stress and increase in relaxation.

Relaxation in the Senior participants saw no significant change from code production to code review even with the decrease in stress. We also saw very similar relaxation levels from all senior participants and Novice group 1 only having slightly elevated levels from their senior counterparts. Novice group 2, again an outlier, had high relaxation levels. The Novice group 1 had a standard deviation in their relaxation scores of 12.7 while Novice 2 had a standard deviation of only 8.76. This could be due to the smaller membership of Novice group 2 but in combination with the other differences exhibited could allude to these participants not taking the tests seriously. Novice group 2 also had the lowest focus and excitement emotions again with a drastic drop in focus and excitement showing little movement. This combination of low excitement low focus and high relaxation in the lowest preforming participants demonstrates a possible behavior of low performing individuals. While Novice group 1 was composed of first year CS students their reviews where generally good with only slightly less detail than their senior counterparts. We also see that the emotion in Novice group 1 and Senior group 1 being within the standard deviation in both stress and excitement. The standard deviation for the other 4 aspects were all lower than 4 points. This tight clustering in emotion with Senior group 1 and Senior group 2 shows that senior level computer science students react similarly to code reviews.

5.3.3 Task 3

Task three introduced a more complex code review for the participants. This code review was a complex code with no style violations. The particular code the participants review was an algorithm for the 8 queens problem. It was markedly harder to understand than the first program, but with the questions outlined in the previous section students where guided on what to think about when reviewing the code.

Table 5.4: Task 3 Group Results

	Novice 1	Novice 2	Senior 1	Senior 2
Engagement	67.35	72.62	55.44	69.55
Excitement	54.09	19.25	43.88	37.51
Focus	58.21	23.64	48.30	47.17
Interest	67.77	61.24	62.25	61.44
Relaxation	36.90	65.60	28.91	25.11
Stress	57.73	65.10	33.88	42.77

This task showed an interesting movement in all the novice groups average emotion aspects. Engagement across both Novice group 1 and Novice group 2 went down only slightly but the other aspects increased. Novice 1 stress moved up by only five points but Novice 2's groups saw it shoot up by over 20 points. Senior group 2 also saw an increase in stress but not as drastic as Novice group 1. This is in contrast to Senior group 1 who saw a large drop in stress. The drop in stress from most of the senior participants could be down to them now getting into better stride of reviewing code but we will see later that there is an interesting pattern when Senior group 1 reviews the complex code. It is also interesting that in the Senior group 1 there was very little deviation in stress with a standard deviation of only 0.15 points.

When looking at Senior group 1's overall emotion aspect we see that every emotion dropped except for interest. This rose by approximately 5 points on average, with a standard deviation of 5.77 points. The notable drops in emotion happen in Engagement, Excitement, Focus and previously mentioned Stress. Some students may have recognized the problem leading them to not need to pay as much attention and become not as stressed while others such as those in Senior 2 not having encountered the problem before. Not all participants had done their undergrad at CalPoly and thus many not have been in a math class where the 8 queens problem was introduced.

Senior group 2's overall emotion from tasks 2 to 3 also saw a large drop in excitement and focus and a moderate drop in engagement. The participants in Senior group 2 all did not fully understand the code for task 3, based on their review comments, but did tie it back to chess. The average interest and relaxation levels for both Senior group 1 and 2 are within 3.5 points of each other. While this code review seemed easy for Senior group 1, with average completion times of 3 minutes 30 seconds, Senior group 2 averaged 5 minutes 20 seconds. This time was also shorter than the novices but Senior group 2's review was much more detailed.

5.3.4 Task 4

Task four starts the section of code reviews that has style violations built in. When looking at the responses from participants about what they wanted to fix, some style recommendations were included but most responses targeted adding more comments or being more clear with what the program was made to do. The program that we had the participants review took in a training data-set and cleaned it for use. We added some unnecessary steps and style violations but the program had no style violations that would prevent it from achieving its purpose.

Table 5.5: Task 4 Group Results

	Novice 1	Novice 2	Senior 1	Senior 2
Engagement	61.75	48.63	59.66	58.51
Excitement	55.85	24.76	36.36	49.62
Focus	54.33	19.62	46.13	56.25
Interest	67.27	64.43	56.47	54.25
Relaxation	33.96	64.21	31.32	23.37
Stress	60.58	54.49	34.62	27.75

The average time to complete the review for this tasks was 3 minutes 10 seconds for Senior group 1, 4 minutes 20 seconds for senior group 2, and approximately 4 minutes for all the novice groups. Most reviewers rated this code relatively high in comparison to the complex code previously encountered. This is due to its ease of understanding despite it having many style violations. Many participants mentioned that it had issues but they would be easily fixed with it getting an average score of 3 compared to the previous codes 2.2 out of 5.

In looking at the code reviews between both senior groups, there was very little difference in their accuracy or quality despite the difference in the emotion. Most of the participants in Senior group 1 focused on changing the loop in the code from being a magic number to a defined variable while those in Senior 2 chose instead to focus on the needCleaning variable and what it was doing.

When looking at the emotion changes from Senior group 1, we see a drop in interest and excitement with Interest level returning to that of task 2 levels and excitement well below that of task 2 levels. This steady decline in excitement leads us to believe that participants in the Senior 1 group may have been experiencing some burnout from doing the code reviews. Senior group 2 on the other hand experienced a large rise in excitement as did Novice group 2. Novice group 1 did not have a significant change in excitement.

While Novice 1 had no drastic changes in emotion from the previous task Novice group 2 did see a huge drop in engagement, dropping on average 24 points. This large drop in engagement did not show any loss in quality from the previous reviews from Novice group 2. This trend will be investigated more fully at the end of section 5.3.6.

This section did see the largest deviation between the members of Senior group 1 with excitement and focus having standard deviations of 20.08 and 20.21 respectfully. The other emotions aspects for this group stayed tightly bound with large swings only in these two emotions. Novice 1 also saw swings in these two emotions with the addition of stress with standard deviations of 11.11 in excitement, 9.98 in focus and 14.96 stress. To put into perspective the average standard deviation in focus for Novice group 1 before this task was only 1.45. This code had some easily recognizable style violations which spread out the emotions of the novices more than that of the seniors.

5.3.5 Task 5

Task five was the last task participants had to do. This task was with the complex code with style violations. The style violations Incorporated in the code were made in such a way as to make them stand out such as using O as a variable name. This code shown in figure **??**. This code is much different than what most participants would be used, to utilizing more data science procedures and nomenclature.

Table 5.6: Task 5 Group Results

	Novice 1	Novice 2	Senior 1	Senior 2
Engagment	61.41	64.33	58.29	56.88
Excitement	51.22	23.66	29.64	52.11
Focus	33.66	21.51	38.51	56.66
Interest	63.14	65.16	57.13	56.55
Relaxation	25.33	62.83	31.13	24.33
Stress	52.16	67.33	29.99	34.66

The average time for all groups increased for this task when compared to the other code review tasks with the novice groups averaging over 8 minutes and the seniors at 6 minutes. This code also incorporates some comments testing their effect on the participants as well as guiding the participants since it is a more difficult snippet of code.

We saw a decline in excitement again for Senior group 2 with like declines in focus and stress. This is in contrast to Senior group 2 whose only significant changes where a slight rise in excitement and stress. The review comments of this code were much more scattered than the previous code reviews with some seniors knowing what was going on and others being completely off. Due to the small group sizes there was no emerging pattern of either group being more or less accurate with this final code review. All senior members mentioned that they liked the comments and the score for this code despite not being understood was significantly higher than the other complex code with an average score of 3.57 out of 5.

This task caused the most deviation in excitement and focus for the novice groups with focus having a standard deviation of 30.64. and excitement 21.41. The other emotion aspects stayed more tightly grouped for Novice group 1 but none of the novices were able to figure out what the code did with many of them wanting to have it explained after the testing concluded. While some novices did try to ask questions during testing, if it pertained to what a section of code did, we would always respond "just try your best". With these questions coming from many of the participants in the novice groups we only saw a rise in engagement from Novice group 2. This would bring Novice group 1 and 2 to have very similar engagement levels but still strongly differing in excitement, relaxation and focus. Novice group 2 and Senior group 2 experienced also a noticeable rise in stress when reviewing this problem that decreased in the other two groups. Overall most of the reviewers appreciated the comments and emotions followed a similar pattern that we will discuss in the next section

5.3.6 Overall Emotion Findings

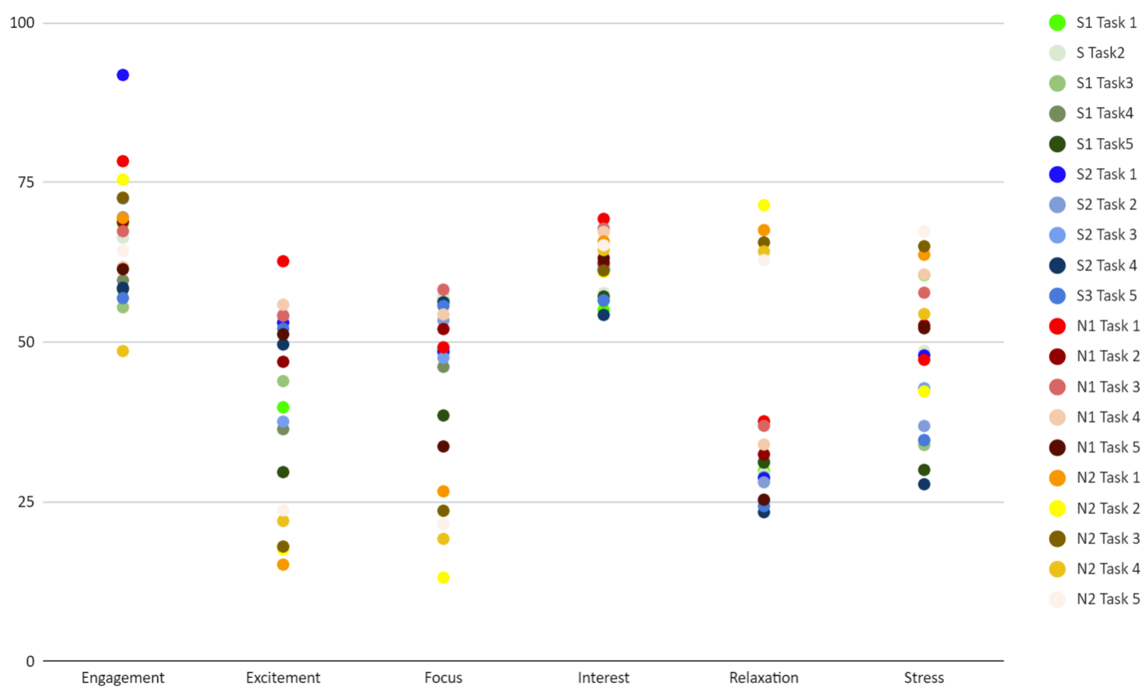

Figure 5.4: Chart of emotion distribution by task and group

When looking at the emotion changes between reviewing code and producing code we see most novices are more engaged and excited with a majority of seniors being more engaged and stressed. These emotion aspects seem to show that most CS students are more engaged when having to produce code with seniors more conscious of trying to do a good job and novices excited to be involved in programming tasks.

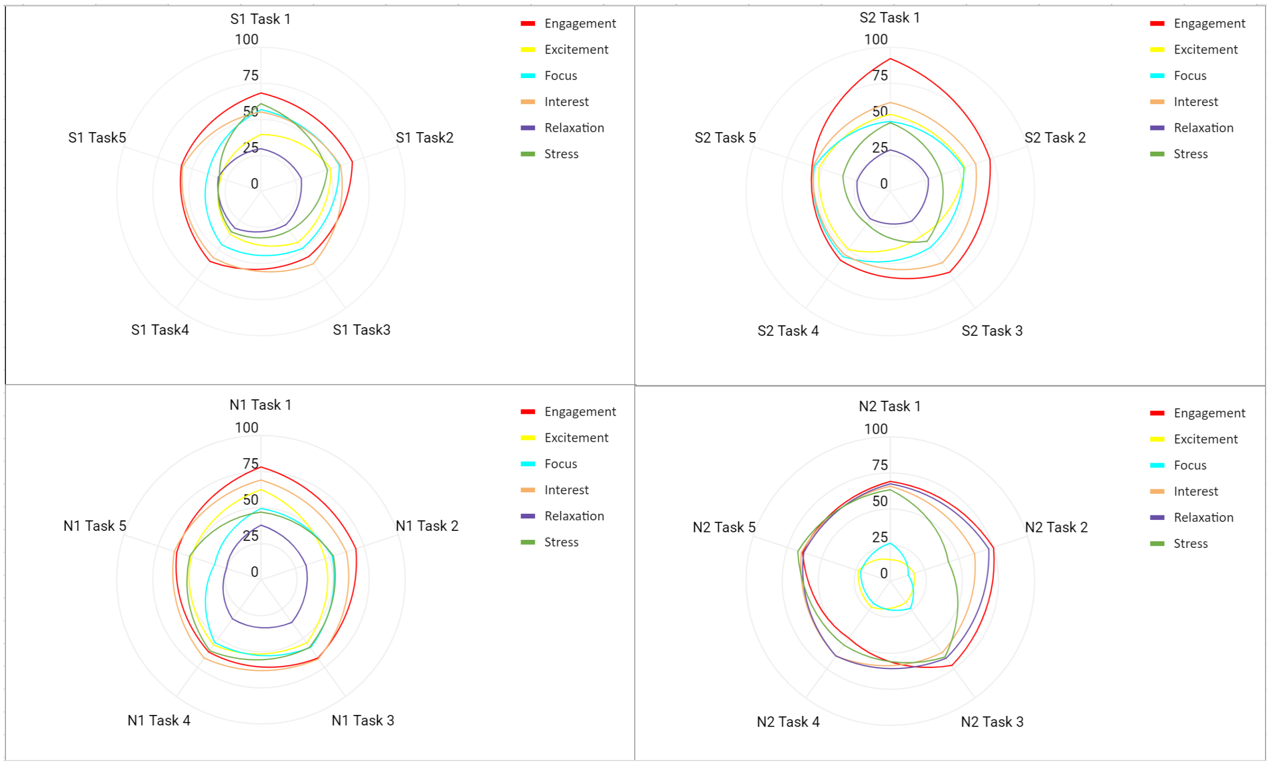

Figure 5.5: Emotion in Radar chart

Senior group 1 was a fairly steady downtrend of excitement continuing from task 2. This along with the slow downtrend of focus and stress shows evidence of burnout. The participants in Senior 1 were less stressed about their tasks while also being less excited and focusing on them. This happened rather quickly as most testing was completed within 40 minutes.

When looking at the comparison of emotion aspects between the two tasks involving complex code other than the downtrends mentioned earlier, the complex code could be seen to expedite certain decreases especially in focus and stress. When conducting these tasks, Senior group 1 had its largest decreases in these categories. This shows that neither style violations nor comments have as big an impact on emotion aspects in code snippets, as the complexity of the code for seniors.

Novice group 1 had its largest drop in focus when encountering the code with comments. This drop did not occur in the first encounter of complex code and appears to be linked with them asking questions, thus getting distracted from their tasks. While it had an effect on focus and relaxation, the other emotion aspect stayed mostly constant.

Overall the Seniors had much less movement in their emotion aspects tending to say close to the mid-line while the novices varied greatly. Most of the seniors deviated only slightly from each other within the same senior group. This could be due to the small sample size or could be evidence of different emotion patterns present in senior students as software developers.

5.4 Conclusion

Although most of the results we achieved were fairly inconclusive we did show that their is a definite change in the emotion aspects between seniors and novices. We showed that more research is warranted in order to understand how both our emotions effect our code and how the code effects us. This research could have a large impact on education and the way we teach intro cs students. By knowing the emotions and cognitive processes that students use when programming and how the code effects these we can assist struggling programmers to learn programming more pleasantly.

Furthermore knowing the best times that we can be productive or how different styles of code affect us is very beneficial. By knowing how the code styles affect us we can write code that would allow us and our colleagues to be the most efficient. We would also know what emotional state allows us to be most productive. We could then code that conjures up those aspects or engage in the proper meditation to put us in the correct state of mind to program.

While the sample size was small due to Covid-19 restrictions there were some interesting trends that were starting to emerge. We were able to answer our research questions and showed that emotion research with EEGs has merit. When looking to the these results there are many avenues that could be pursued from the findings, that we will explore in our future works.

5.4.1 Revisiting the Research Questions

With the amount of participants that we were able to achieve we were not able to definitively solve all of our research questions but did make good steps in answering them. Using the K-means algorithm to group the participants show that we did have groups emerging in the different data sets. This also showed that the novices where much more scattered in their measured emotion aspects then the seniors.

Answering our first research question: Using a portable EEG can we find an association between the emotions the developers feel and their correctness and experience when performing software engineering tasks? With just the k-means algorithm analyzing the emotion data we could differentiate the seniors from the novices. This trend does seem to allude to an answer to our first research question. If the trend observed in our small sample holds true for a larger data-set or becomes more apparent we would be able to associate the emotions that developers feel with their experience and accuracy when preforming tasks. We showed one group of seniors who performed not as well as another with a difference in emotion aspects. While Novice group 2 did not perform as well as Novice group 1 its membership was quite small and neither were very cohesive as a group. Overall the differences inside the senior group was much smaller than its difference from the novice groups.

When analyzing our second research question: What effect does code style have on developers' emotions and their level of comprehension while having to read and interpret code? We did not see a drastic change in emotion given different styles. Instead what we saw was that the code complexity had a much large impact on software engineers. Another factor that had a big impact on software engineers is the length of time they are preforming tasks with out participants appearing to get burnt out rather quickly. This as mentioned in future works should still be investigated more thoroughly with a larger sample size of experienced software engineers.

5.5 Future work

With the small sample size and recruitment pool limited to students and faculty it would be very beneficial to run another study using developers in industry. This would allow us to see how software engineers with more practical experience express their emotions when programming. It would also be beneficial to have separate code samples for experienced programmers to review allowing for interesting style errors and the effect of code structure on the emotion of developers.

A separate study investigating the changes of students as they matriculate through the software engineer academic pipeline could also be conducted. This would be more in line with a longitudinal study investigating how the novices develop and change after becoming more experienced. Since investigators would have the same participants to compare against this could potentially be done with less participants.

Overall, EEG research has shown to be very easy and noninvasive to pursue. Some techniques such as extracting the brain visualizations need to be better explored but the base ground work is there. If we can mimic many of the functions of an fMRI with something as cheap and portable as an EEG then this type of research will be more accessible to less funded research groups.

BIBLIOGRAPHY

[1] M. I. Al-Kadi, M. B. I. Reaz, and M. A. M. Ali. Evolution of electroencephalogram signal analysis techniques during anesthesia. *Sensors*, 13(5):6605–6635, 2013.

[2] D. Alessia, F. Ferri, P. Grifoni, and T. Guzzo. Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, 125(3), 2015.

[3] S. S. Bae and M. E. West. Cyborg crafts: Second skin (soft keen interaction). In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 1–3, 2021.

[4] J. W. Britton, L. C. Frey, J. L. Hopp, P. Korb, M. Z. Koubeissi, W. E. Lievens, E. M. Pestana-Knight, and E. K. St Louis. Electroencephalography (eeg): an introductory text and atlas of normal and abnormal findings in adults, children, and infants. 2016.

[5] C. Casalnuovo, K. Lee, H. Wang, P. Devambu, and E. Morgan. Do programmers prefer predictable expressions in code? *Cognitive Science*, 44(12), 2020.

[6] M. Chaudhary, S. Mukhopadhyay, M. Litoiu, L. E. Sergio, and M. S. Adams. Understanding brain dynamics for color perception using wearable eeg headband. *arXiv preprint arXiv:2008.07092*, 2020.

[7] G. L. Clore and J. R. Huntsinger. How emotions inform judgment and regulate thought. *Trends in cognitive sciences*, 11(9):393–399, 2007.

[8] I. A. Cook, R. O'Hara, S. H. Uijtdehaage, M. Mandelkern, and A. F. Leuchter. Assessing the accuracy of topographic eeg mapping for determining local brain function. *Electroencephalography and clinical neurophysiology*, 107(6):408–414, 1998.

[9] Emotiv epoc 14-channel wireless eeg headset, Sep 2020.

[10] W. Fei, L. Bi, and J. Zhang. Adaptive brain-machine interface of brain-controlled vehicles using semi-mim and tsvm. In *Proceedings of the 5th International Conference on Control Engineering and Artificial Intelligence*, pages 31–35, 2021.

[11] A. Ferng. Brodmann areas, Oct 2020.

[12] R. S. Frackowiak. *Human brain function*. Elsevier, 2004.

[13] D. Fucci, D. Girardi, N. Novielli, L. Quaranta, and F. Lanubile. A replication study on code comprehension and expertise using lightweight biometric sensors. In *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, pages 311–322. IEEE, 2019.

[14] C. Gifford. The problem with emotion-detection technology.

[15] D. Girardi, F. Lanubile, and N. Novielli. Emotion detection using noninvasive low cost sensors. In *2017 Seventh international conference on affective computing and intelligent interaction (ACII)*, pages 125–130. IEEE, 2017.

[16] D. Girardi, F. Lanubile, N. Novielli, and D. Fucci. Sensing developers' emotions: The design of a replicated experiment. In *2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, pages 51–54. IEEE, 2018.

[17] D. Girardi, N. Novielli, D. Fucci, and F. Lanubile. Recognizing developers' emotions while programming. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 666–677, 2020.

[18] L. J. Gonçales, K. Farias, and B. C. da Silva. Measuring the cognitive load of software developers: An extended systematic mapping study. *Information and Software Technology*, 136:106563, 2021.

[19] Google.

[20] R. E. Gur, C. McGrath, R. M. Chan, L. Schroeder, T. Turner, B. I. Turetsky, C. Kohler, D. Alsop, J. Maldjian, J. D. Ragland, et al. An fmri study of facial emotion processing in patients with schizophrenia. *American Journal of Psychiatry*, 159(12):1992–1999, 2002.

[21] D. Himmelman and J. Chu. A measurement of alertness based on electroencephalographic time series analysis. In *Proceedings of the 1961 16th ACM national meeting*, pages 133–101, 1961.

[22] S. Jung, R. Xiao, O. Buruk, and J. Hamari. Designing gaming wearables: From participatory design to concept creation. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 1–14, 2021.

[23] V. Jurcak, D. Tsuzuki, and I. Dan. 10/20, 10/10, and 10/5 systems revisited: their validity as relative head-surface-based positioning systems. *Neuroimage*, 34(4):1600–1611, 2007.

[24] M. L. Kesler/West, A. H. Andersen, C. D. Smith, M. J. Avison, C. Davis, R. J. Kryscio, and L. X. Blonder. Neural substrates of facial emotion processing using fmri. *Cognitive Brain Research*, 11(2):213–226, 2001.

[25] J. L. Lobo, J. D. Ser, F. De Simone, R. Presta, S. Collina, and Z. Moravek. Cognitive workload classification using eye-tracking and eeg data. In *Proceedings of the International Conference on Human-Computer Interaction in Aerospace*, pages 1–8, 2016.

[26] Mayfield brain & spine.

[27] C. M. Michel and M. M. Murray. Towards the utilization of eeg as a brain imaging tool. *Neuroimage*, 61(2):371–385, 2012.

[28] S. C. Müller and T. Fritz. Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 688–699. IEEE, 2015.

[29] S. C. Müller and T. Fritz. Using (bio)metrics to predict code quality online. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pages 452–463, 2016.

[30] Y. Nakagawa, P. Sripian, and M. Sugaya. Evaluation of distance learning on concentration and relax by eeg and hrv. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 762–763, 2020.

[31] N. Novielli, F. Calefato, D. Dongiovanni, D. Girardi, and F. Lanubile. Can we use se-specific sentiment analysis tools in a cross-platform setting? In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 158–168, 2020.

[32] S. of Queensland. Brain map, 2013.

[33] D. Ringach. The limits of fmri, Jul 2009.

[34] A. Scarantino and R. de Sousa. Emotion. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2021 edition, 2021.

[35] E. Schneiders, M. B. Kristensen, M. K. Svangren, and M. B. Skov. Temporal impact on cognitive distraction detection for car drivers using eeg. In *32nd Australian Conference on Human-Computer Interaction*, pages 594–601, 2020.

[36] N. Schwarz. Emotion, cognition, and decision making. *Cognition & Emotion*, 14(4):433–440, 2000.

[37] T. M. Shaft. Helping programmers understand computer programs: the use of metacognition. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 26(4):25–46, 1995.

[38] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th international conference on software engineering*, pages 378–389, 2014.

[39] J. Siegmund, N. Peitek, C. Parnin, S. Apel, J. Hofmeister, C. Kästner, A. Begel, A. Bethmann, and A. Brechmann. Measuring neural efficiency of program comprehension. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 140–150, 2017.

[40] The difference between an expert's brain and a novice's, Nov 2019.

[41] R. D. Vieira and K. Farias. Usage of psychophysiological data as an improvement in the context of software engineering: A systematic mapping study. In *XVI Brazilian Symposium on Information Systems*, pages 1–8, 2020.

[42] A. Vourvopoulos, E. Niforatos, S. Bermudez i Badia, and F. Liarokapis. Brain–computer interfacing with interactive systems—case study 2. In *Intelligent Computing for Interactive System Design: Statistics, Digital Signal Processing, and Machine Learning in Practice*, pages 237–272. 2021.

[43] What is fmri?

[44] Z. Wimshurst, P. Sowden, and M. Wright. Expert–novice differences in brain function of field hockey players. *Neuroscience*, 315:31–44, 2016.

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

[46] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo, and L. Jiang. Sentiment analysis for software engineering: How far can pre-trained transformer models go? In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 70–80, 2020.