Investigating Cognitive Demands in Software Testing Activities: An EEG-Based Experimental Study

Roselane Silva Farias
Institute of Computing (IC)
Federal University of Bahia
Salvador, Brazil
silva.roselane@ufba.br

Dhyego Tavares
Institute of Computing (IC)
Federal University of Bahia
Salvador, Brazil
dhyegocruz@ufba.br

José Garcia Vivas Miranda
Institute of Physics
Federal University of Bahia
Salvador, Brazil
vivas@ufba.br

Eduardo Santana de Almeida
Institute of Computing (IC)
Federal University of Bahia
Salvador, Brazil
eduardo.almeida@ufba.br

Abstract

Software testing encompasses a variety of cognitively demanding activities, including code comprehension, fault identification, and test case design. Understanding the cognitive demands associated with these activities is important for improving testing processes, training programs, and tool support. However, despite the central role of software testers in software quality assurance, the cognitive processes underlying testing activities remain relatively underexplored. In this paper, we investigate neural activity and performance during software testing using electroencephalography (EEG). We conducted a controlled experiment with eight professional software testers performing four common testing activities: test code comprehension, syntax error identification, logic error identification, and test case design. Our results provide descriptive evidence of variation in neural activity across software testing activities, although the Friedman tests did not reveal statistically significant differences among activities. Logic Error Identification produced the largest reduction in Alpha activity and the lowest participant accuracy, suggesting greater attentional and analytical demands. Test Case Design, meanwhile, exhibited the highest Theta and Delta activity levels while maintaining relatively high performance, suggesting greater engagement of cognitive resources, including working memory. Furthermore, a strong positive association was observed between Theta activity and performance during Logic Error Identification ($\rho = 0.77$, $p = 0.025$). Overall, these findings provide preliminary evidence that software testing activities may involve different cognitive demands, motivating further investigation into how cognitive insights can inform testing practices, training strategies, automation techniques, and AI-assisted testing tools.

Keywords

Software Testing, Cognitive Load, Electroencephalography (EEG), Human Factors, Software Engineering

1 Introduction

Software testing is a cornerstone of software engineering, ensuring that software systems meet the highest standards of quality, reliability, and security [18]. Modern testing activities encompass a broad range of tasks, including test automation, fault detection, debugging, and test case design, each requiring different forms of reasoning and decision-making [20]. The effectiveness of these activities directly impacts software quality, development costs, and the successful adoption of testing tools and automation techniques.

Despite significant advances in testing processes, automation frameworks, and AI-assisted testing approaches, software testing remains a highly human-centered activity. Testers are constantly required to comprehend test artifacts, identify faults, reason about system behavior, and design effective test cases [9, 18, 26]. These activities impose different cognitive demands that may influence testing effectiveness and productivity. However, little is known about how software testers cognitively process these activities and which testing tasks impose the greatest mental effort.

Understanding the cognitive demands associated with software testing activities can provide valuable insights for improving tester training, designing more effective testing processes, and developing intelligent tools that better support testers during cognitively demanding tasks.

Previous studies have examined the mental and neural demands of programming-related tasks, including code comprehension, debugging, and algorithmic reasoning [4, 15, 21, 37]. These studies reveal that such activities engage distinct cognitive processes that vary across individuals and experience levels. Experienced developers, for instance, tend to exhibit enhanced memory activation [6, 12, 16, 33] and reduced cognitive workload [6]. However, software testing, despite being equally complex and cognitively demanding, has received comparatively little attention from the neurocognitive perspective.

Testing differs fundamentally from development. While development emphasizes constructive reasoning and solution creation, testing centers on evaluative reasoning, fault detection, and hypothesis-driven exploration [10, 34]. These cognitive distinctions are likely to engage different neural mechanisms, potentially resulting in distinct patterns of brain activation. Yet, to date, these differences have not been systematically explored.

This distinction is particularly relevant as modern software testing increasingly relies on automation frameworks and AI-assisted tools, whose effectiveness depends on understanding the challenges faced by testers during fault detection and test design activities.

Recent advances in neuroimaging provide promising tools for bridging this gap. Techniques such as electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) have long been employed to investigate cognitive states and brain activity in complex problem-solving tasks [35]. Applying these methods to software testing offers a novel opportunity to explore the cognitive underpinnings of evaluative reasoning in software engineering.

In this study, we investigate the cognitive demands associated with different software testing activities through electroencephalography (EEG). Our goal is not only to characterize neural activation patterns during testing tasks, but also to better understand which testing activities impose higher cognitive load and may therefore benefit from improved processes, training strategies, and tool support. To this end, we conducted a controlled experiment with eight professional software testers performing four common testing activities: test automation code comprehension, syntax error identification, logic error identification, and test case design.

To better understand the cognitive demands associated with software testing activities, this study addresses the following research questions:

RQ1: How do different software testing activities differ in terms of neural activation? Different testing activities require distinct cognitive processes, ranging from code comprehension and fault detection to test case design. Understanding whether these activities show different patterns of neural activation may provide exploratory insights into their cognitive demands.

RQ2: Which testing activities impose higher cognitive load on software testers?

Understanding which activities impose higher cognitive load can help identify tasks that are more mentally demanding for testers and therefore may benefit from improved training or process support. Such insights may contribute to improving both tester effectiveness and software quality.

RQ3: Is cognitive load associated with testing performance?

Understanding whether cognitively demanding activities are also associated with lower task performance may help identify testing activities that would benefit most from improved tool support, automation, or AI-assisted solutions.

Our results provide preliminary descriptive evidence of variation in neural activity and performance across testing activities, with fault identification and test case design showing comparatively stronger indicators of cognitive effort.

In summary, this work contributes to software testing research in three main ways:

- We provide the first empirical EEG-based investigation of cognitive demands during software testing activities.
- We characterize the cognitive demands of four common testing activities, including test code comprehension, syntax error identification, logic error identification, and test case design.
- We publicly release all experimental materials, tasks, and data to support replication and future research on human factors in software testing.

2 Background and Related Work

Software quality assurance (QA) aims to ensure that software engineering processes, methodologies, and deliverables comply with established standards and requirements, ultimately contributing to software quality, reliability, and safety [25]. Within QA, software testing plays a central role in identifying defects, validating system behavior, and ensuring that software meets functional and non-functional requirements.

Software defects remain a persistent challenge for both industry and society, generating substantial economic and operational costs [19]. Consequently, improving the effectiveness of software testing activities is an important research objective. Although significant advances have been achieved in testing methodologies, automation frameworks, and AI-assisted testing approaches, software testing remains a highly human-centered activity. Testers must comprehend testing artifacts, identify defects, evaluate system behavior, and design effective test cases. Understanding the cognitive demands associated with these activities may contribute to improved testing processes, training strategies, and tool support.

Recent advances in neuroscience and neuroimaging have opened new opportunities for investigating cognitive processes in software engineering. Techniques such as electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) have been increasingly employed to study how software professionals perform cognitively demanding tasks [14, 23, 24, 30, 31, 33, 35, 37]. While fMRI provides high spatial resolution and enables precise localization of brain activity, its high cost and operational constraints often limit its applicability in software engineering research.

In contrast, EEG offers a more practical and cost-effective alternative for studying software engineering activities. Besides being non-invasive and relatively inexpensive, EEG provides high temporal resolution, allowing researchers to capture rapid changes in brain activity during task execution [35]. These characteristics make EEG particularly suitable for investigating cognitive load, attention, and mental effort during software-related activities.

The remainder of this section is organized as follows. First, we provide a brief overview of EEG and its main characteristics. Next, we review previous neuroimaging studies in software engineering and discuss the research gap that motivates this work.

2.1 Electroencephalogram (EEG)

The first human electroencephalogram (EEG) recording was performed by Hans Berger in 1924 [27]. Since then, EEG has become one of the most widely adopted tools for investigating cognitive processes in neuroscience and, more recently, software engineering research [6, 15, 16, 22].

EEG measures voltage fluctuations generated by ionic currents associated with neuronal activity in the cerebral cortex [27]. Using electrodes positioned on the scalp, EEG captures synchronized postsynaptic activity from large populations of neurons with millisecond-level temporal precision [35]. This temporal resolution makes EEG particularly suitable for studying dynamic cognitive processes during software-related tasks.

The EEG signal consists of multiple oscillatory wave patterns, commonly categorized into four primary frequency bands: Delta, Theta, Alpha, and Beta. These bands have been associated with different cognitive and physiological states, as summarized in Table 1. In cognitive tasks, Alpha activity is often associated with relaxed mental states and tends to decrease as cognitive demand increases, whereas Theta activity is frequently linked to working memory and cognitive processing [13]. These characteristics make EEG particularly useful for investigating cognitive load and mental effort during software engineering activities.

Table 1: EEG Frequencies and Associated Cognitive States

EEG Frequency	Associated Cognitive State
Delta (0.5–4 Hz)	Deep sleep and low-arousal states
Theta (4–8 Hz)	Working memory and cognitive processing
Alpha (8–13 Hz)	Relaxed state; decreases with cognitive effort
Beta (13–30 Hz)	Attention and active cognitive processing

Understanding the cognitive demands associated with software engineering activities has become an important research topic. Neuroimaging studies have provided valuable insights into factors such as cognitive load, attention, expertise, and task performance during software development activities. However, relatively little is known about the cognitive processes involved in software testing, despite its central role in software quality assurance.

Most neuroimaging studies in software engineering have focused on programming-oriented tasks, including code comprehension, debugging, and code review [6–8, 12, 16, 22, 38]. These studies demonstrate that neurophysiological measurements can reveal meaningful differences in cognitive workload and performance during software development activities.

Calcagno et al. [4] used EEG to investigate software development tasks and observed increases in Delta and Theta activity associated with working memory and mental workload, as well as elevated Beta activity related to attention and task engagement.

Lin et al. [16] combined EEG and eye tracking to compare novice and experienced programmers. Their results suggest that high-performing programmers exhibit stronger working memory activation and attentional control, reflected in distinct Theta and Alpha activity patterns.

Similarly, Peitek et al. [22] analyzed EEG and eye-tracking data from programmers performing code comprehension tasks. They found that more effective programmers exhibited lower cognitive load, reduced Beta activity, and increased Alpha activity during task execution.

Taken together, these studies consistently demonstrate that software-related activities impose measurable cognitive demands and that neurophysiological signals can reveal differences associated with expertise, task complexity, cognitive workload, and performance. However, the existing body of evidence remains concentrated on programming-oriented activities.

Software testing differs fundamentally from software development. While programming activities primarily focus on solution construction, software testing emphasizes fault detection, validation, and hypothesis-driven reasoning [10, 34]. Testers must continuously evaluate system behavior, identify defects, analyze testing artifacts, and design effective test cases. These activities involve cognitive processes that may differ substantially from those observed during code comprehension and software development tasks.

Despite the growing body of neuroimaging research in software engineering, software testing remains largely unexplored from a neurocognitive perspective. As modern testing increasingly relies on automation frameworks, continuous integration pipelines, and AI-assisted testing tools, understanding the cognitive demands associated with testing activities becomes particularly relevant. Such insights may help identify testing activities that would benefit from improved training, enhanced tool support, or additional automation.

To the best of our knowledge, no prior EEG study has specifically investigated the cognitive processes associated with common software testing activities performed by professional software testers. This study addresses this gap by examining neural activation and cognitive load during test automation code comprehension, syntax error identification, logic error identification, and test case design activities.

3 Experimental Study

In this section, we describe the participants, experimental design, software testing tasks, EEG acquisition procedures, and statistical analyses used to investigate the cognitive demands associated with different software testing activities.

3.1 Participants

We conducted a controlled experiment with eight software testing professionals who performed software testing tasks while their brain activity was recorded using an EEG. The study was approved by the Research Ethics Committee of the School of Nursing, Federal University of Bahia (UFBA), Bahia, Brazil, in May 2024 (Approval No. 78645624.2.0000.5531).

Participants were recruited through convenience sampling [36] from local software companies. All participants had prior experience with software testing and test automation using Java, ensuring familiarity with the types of activities performed during the experiment. This selection criterion was adopted to increase the ecological validity of the study and ensure that the tasks reflected activities commonly encountered in industrial software testing practice.

Table 2 summarizes the demographic characteristics of the participants.

Table 2: Demographic Information of the Participants

Gender		Expertise Level		
Men	Women	Novice	Intermediate	Advanced
5	3	2	3	3

3.1.1 Exclusion Criteria. Exclusion criteria for this study included individuals using psychotropic drugs or medications that could impact brain function. Additionally, participants with cognitive deficits, visual impairments, or any other medical condition listed in the pre-experiment medical form were excluded.

3.2 Experimental Design

The experiment consisted of a baseline condition followed by four software testing task blocks. The baseline condition served as a reference for identifying task-related changes in neural activation and cognitive load during testing activities.

During the baseline condition, participants remained seated with their eyes open for three minutes while focusing on a fixation cross displayed on the screen. This procedure is commonly adopted in EEG studies to establish a stable neural reference condition.

Following the baseline period, participants completed four blocks of software testing activities. Task completion time varied across participants. Figure 1 illustrates the overall experimental protocol.

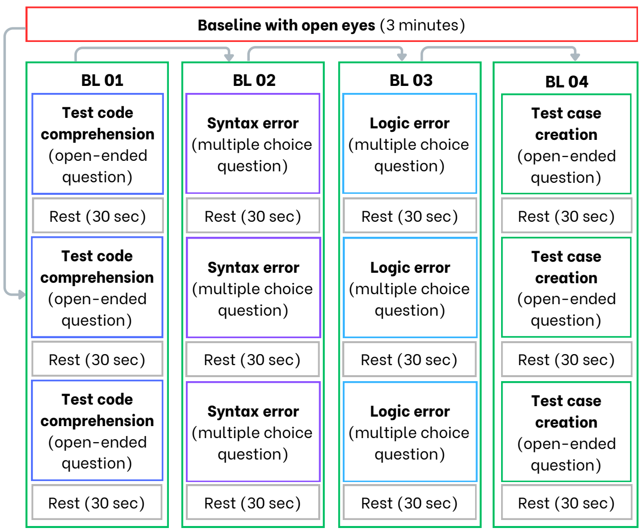

Figure 1: Visualization of the experiment design, which shows four blocks of tasks (BL01–BL04).

3.3 Experimental Tool

We used PROPHET, a Java-based infrastructure for software engineering experiments that supports task presentation and response collection [32]. The source code is publicly available at https://github.com/feigensp/Prophet.

3.3.1 Programming Language. Java was selected because it is widely used in software testing and test automation practice [2] and is frequently adopted in software engineering experiments involving code comprehension and maintenance tasks [3, 21, 31, 33, 37].

3.4 Testing Tasks

The selected tasks were designed to represent common activities performed by software testers during test automation and quality assurance processes. Together, they cover test code comprehension, fault identification, and test case design activities that are frequently encountered in industrial software testing practice.

The tasks were refined through pilot studies conducted prior to the experiment to ensure appropriate difficulty and clarity. All task materials are available in the replication package.

- **BL01 – Test Code Comprehension:** Participants analyzed test automation code snippets and answered open-ended questions regarding their behavior and expected outputs.
- **BL02 – Syntax Error Identification:** Participants inspected test automation code and identified syntactic defects through multiple-choice questions.
- **BL03 – Logic Error Identification:** Participants analyzed test automation code and identified behavioral defects affecting test execution.
- **BL04 – Test Case Design:** Participants designed and extended test cases based on software requirements and expected system behavior.

Together, these four task categories capture complementary aspects of software testing activities, enabling the investigation of cognitive demands associated with code comprehension, fault detection, and test design.

3.5 EEG Acquisition and Analysis

Brain electrical activity was recorded using a 64-channel EEG system following the international 10–10 electrode placement standard. The reference electrode was positioned at Cz, and electrode-skin impedance was maintained below 50 kΩ throughout data acquisition. Signals were sampled at 1000 Hz to ensure high temporal resolution.

EEG preprocessing and analysis were performed using EEGLAB in MATLAB R2018b. Raw signals were filtered using a band-pass filter between 0.5 and 50 Hz to remove low-frequency drifts and high-frequency noise. The continuous EEG recordings were initially segmented into 1.5-second epochs during preprocessing to facilitate artifact detection and signal inspection.

To improve data quality, Independent Component Analysis (ICA) was applied to automatically identify and remove ocular, electromyographic, and electrocardiographic artifacts.[1] Additionally, epochs containing signal amplitudes exceeding $\pm 100\ \mu V$ were automatically rejected, as such amplitudes typically indicate excessive noise or movement-related artifacts [5]. A subsequent manual visual inspection was performed to identify and remove any remaining artifacts.

Following preprocessing, quantitative EEG (qEEG) analysis was conducted to characterize neural activity across different frequency bands [17]. Using custom MATLAB scripts, power spectral density (PSD) analysis was performed on artifact-free 1-second non-overlapping epochs. Welch's method was employed to estimate the average PSD for each participant and task condition.

The analysis focused on four frequency bands commonly associated with cognitive processing: Delta (1–4 Hz), Theta (4–8 Hz), Alpha (8–13 Hz), and Beta (13–30 Hz). For each frequency band, absolute power values were computed to quantify neural activation during the baseline condition and software testing activities.

Figure 2 summarizes the EEG processing pipeline, including data acquisition using Profusion software, preprocessing procedures conducted in MATLAB, and the subsequent statistical analyses performed in SPSS.

3.6 Statistical Analysis

Before performing the statistical analyses, data normality was assessed using the Shapiro–Wilk test [28]. Given the small sample size ($N = 8$) and the non-normality observed in several EEG variables, non-parametric statistical methods were used throughout the analysis [29].

[1]Artifacts correspond to non-neural signals that may contaminate EEG recordings.

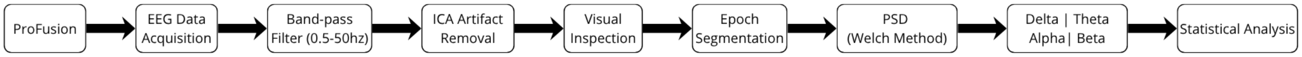

Figure 2: Overview of the EEG processing pipeline used in this study, including signal preprocessing, artifact removal, epoch segmentation, power spectral density analysis, frequency band extraction, and statistical analysis.

To investigate whether neural activation differed across the four testing activities (BL01–BL04), we employed the *Friedman* test, a non-parametric alternative to repeated-measures ANOVA suitable for within-subject experimental designs. The Friedman test was applied independently to each EEG frequency band (Alpha, Beta, Theta, and Delta), accounting for repeated measurements from each participant.

Effect sizes were quantified using Kendall's coefficient of concordance (W), which provides an estimate of the magnitude of differences among repeated measurements independently of statistical significance.

Descriptive analyses were additionally conducted by comparing the average EEG activity observed during each testing activity with the corresponding baseline condition. These baseline differences were used to facilitate the interpretation of the neural activation patterns but were not subjected to inferential statistical testing. All statistical analyses adopted a significance level of $\alpha = 0.05$, and hypothesis tests were evaluated using two-tailed p-values.

Given the exploratory nature of this study and the challenges associated with recruiting experienced software testing professionals for EEG experiments, the findings should be interpreted as preliminary evidence rather than definitive conclusions. The repeated-measures experimental design mitigates inter-participant variability; however, the limited sample size reduces the statistical power to detect small effects. Consequently, the reported results should be viewed as an initial characterization of the neural activity associated with software testing activities, motivating future confirmatory studies involving larger and more diverse participant samples.

4 Results

This section presents the results of the statistical analyses performed to investigate neural activation, cognitive load, and task performance during software testing activities. Following the research questions introduced in Section 1, we first analyze differences in neural activation across testing activities (RQ1), then identify which activities impose higher cognitive load (RQ2), and finally examine whether cognitive load is associated with task performance (RQ3).

4.1 RQ1: How Do Different Software Testing Activities Differ in Terms of Neural Activation?

To investigate differences in neural activation across software testing activities, we analyzed the average EEG power in the Alpha, Beta, Theta, and Delta frequency bands within the frontal brain region. This region was selected because it is strongly associated with executive cognitive functions, including reasoning, decision-making, and working memory [13].

Figure 3 presents the average EEG Power Spectral Density (PSD) observed across all participants for each testing activity and frequency band. Warmer colors indicate higher activity, whereas cooler colors indicate lower activity, allowing direct comparison of neural activation patterns across tasks.

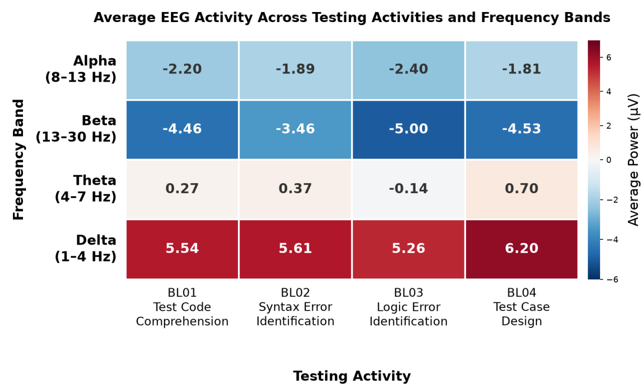

Figure 3: Average frontal EEG power across software testing activities and frequency bands.

Distinct activation profiles emerged across the evaluated activities. Test Case Design (BL04) exhibited the highest Theta (0.70) and Delta (6.20) activity levels, while Syntax Error Identification (BL02) showed the highest Beta activity (-3.46). Logic Error Identification (BL03) presented the lowest Alpha activity (-2.40), suggesting a different neural activation pattern compared to the remaining tasks.

To better understand task-related changes, Table 3 presents the average deviation from the baseline condition.

Table 3: Average deviation from baseline EEG activity.

Band	BL01	BL02	BL03	BL04
Alpha	-0.36	-0.05	-0.56	+0.03
Beta	+1.89	+2.89	+1.35	+1.82
Theta	+0.63	+0.73	+0.22	+1.06
Delta	+0.61	+0.68	+0.33	+1.27

All testing activities produced increases in Beta, Theta, and Delta activity relative to the baseline condition. The largest deviations were observed during Test Case Design (BL04), particularly for Theta (+1.06) and Delta (+1.27). Syntax Error Identification (BL02) produced the strongest increase in Beta activity (+2.89), whereas Logic Error Identification (BL03) exhibited the largest reduction in Alpha activity (-0.56).

To assess whether the observed activation differences were statistically significant, we performed *Friedman* tests across the four testing activities for each EEG frequency band. Table 4 summarizes the results.

Table 4: Friedman test results comparing EEG activity across testing activities.

Band	$\chi^2(3)$	p	Kendall's W
Alpha	1.05	0.789	0.044
Beta	1.05	0.789	0.044
Theta	4.95	0.175	0.206
Delta	4.05	0.256	0.169

The *Friedman* tests did not reveal statistically significant differences among the four testing activities for any of the analyzed frequency bands ($p > 0.05$). These results indicate that, within the studied sample, the observed variations in EEG activity cannot be considered statistically distinguishable across activities.

Nevertheless, differences in effect size were observed across frequency bands. Theta activity presented the largest effect size (Kendall's $W = 0.206$), followed by Delta activity ($W = 0.169$). Although these effect sizes are small, they suggest that task-related variations may be more pronounced in the lower-frequency bands, particularly Theta, which has often been associated with working-memory and cognitive-control processes.

RQ1 Summary. Descriptive variation in EEG activity was observed across the evaluated software testing activities. Test Case Design (BL04) exhibited the highest Theta and Delta activity levels, whereas Logic Error Identification (BL03) showed the lowest Alpha activity.
Although the *Friedman* tests did not reveal statistically significant differences among activities in any EEG frequency band, Theta exhibited the largest observed effect size ($W = 0.206$). These results should be interpreted as exploratory patterns rather than evidence of statistically established differences in neural activation.

4.2 RQ2: Which Testing Activities Impose Higher Cognitive Load on Software Testers?

According to Antonenko et al., mental effort is a fundamental component of cognitive load, as it reflects the cognitive resources allocated to addressing task demands [1]. To investigate cognitive load during software testing activities, we analyzed changes in EEG activity relative to the baseline condition, focusing primarily on the Alpha and Theta frequency bands.

Prior EEG research associates reductions in Alpha activity with increased cognitive effort and attentional demands, whereas increases in Theta activity are commonly linked to working-memory engagement and mental workload [11, 13]. Therefore, deviations in these frequency bands may serve as useful indicators of the cognitive demands imposed by different testing activities.

To interpret the observed activation patterns in terms of cognitive load, we focused on deviations from the baseline condition in the Alpha and Theta bands, which are widely associated with attentional effort and working-memory demands, respectively.

Among the evaluated activities in Table 3, Logic Error Identification (BL03) produced the strongest Alpha suppression relative to baseline, whereas Test Case Design (BL04) generated the largest increase in Theta activity.

In contrast, Test Code Comprehension (BL01) exhibited the lowest Beta increase among all activities (+1.89), while achieving the highest accuracy, suggesting that participants were generally able to perform this task with comparatively lower active cognitive processing demands. Syntax Error Identification (BL02) produced the strongest increase in Beta activity relative to baseline (+2.89).

Theta activity has frequently been associated with working-memory utilization and mental workload in neuroscience studies [13]. Therefore, the elevated Theta activity observed during BL04 (+1.06) — the highest among all activities — may indicate greater engagement of cognitive resources related to planning, reasoning, and decision-making. Notably, BL01 and BL02 also showed positive Theta deviations (+0.63 and +0.73, respectively), while BL03 presented the lowest Theta increase (+0.22), suggesting that working-memory demands were comparatively lower during logic error identification than during the remaining activities.

From a cognitive perspective, these patterns may reflect different forms of mental demand. The Alpha suppression observed during BL03 may indicate increased attentional effort and analytical reasoning associated with fault localization and behavioral analysis, consistent with the well-established relationship between Alpha decreases and cognitive effort (Table 3). In contrast, the increased Theta and Delta activity observed during BL04 may reflect greater working-memory engagement required to design test cases, reason about alternative execution scenarios, and anticipate system behavior.

When these neural findings are considered together with the performance results presented in RQ3, an interesting pattern emerges. BL03 exhibited both the strongest Alpha suppression and the lowest participant accuracy, suggesting comparatively greater attentional demands in this sample. Conversely, BL04 generated the strongest overall Theta and Delta activation while maintaining relatively high performance levels, indicating that cognitively demanding activities, particularly those requiring planning and working-memory engagement, do not necessarily lead to poorer outcomes.

RQ2 Summary. Logic Error Identification (BL03) and Test Case Design (BL04) exhibited comparatively stronger descriptive indicators of cognitive effort. BL03 showed the largest reduction in Alpha activity relative to baseline, while BL04 showed the largest increases in Theta and Delta activity. These patterns may indicate greater attentional and working-memory demands, respectively, but should be interpreted cautiously given the small sample size.

4.3 RQ3: Is Cognitive Load Associated With Testing Performance?

Participant performance was calculated as the percentage of correctly answered questions within each activity block. Figure 4 presents the average accuracy achieved by participants across the evaluated testing activities.

Substantial performance differences were observed among activities. Test Code Comprehension (BL01) achieved the highest average accuracy (79.17%), followed by Test Case Design (BL04) with 70.83%.

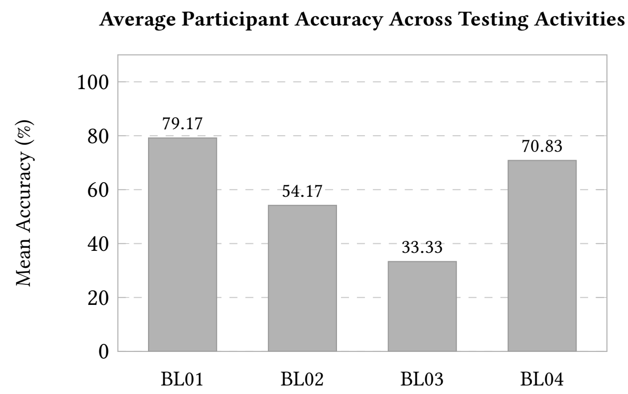

Figure 4: Average participant accuracy across the four testing activities.

Syntax Error Identification (BL02) produced intermediate performance (54.17%), whereas Logic Error Identification (BL03) yielded the lowest accuracy (33.33%).

These results suggest that activities involving fault identification, particularly behavioral fault detection, showed lower observed accuracy than code comprehension and test case design tasks.

To further explore the relationship between neural activity and performance, we computed *Spearman*'s rank correlations between EEG activity and participant accuracy. A strong positive association was observed between Theta activity and performance during BL03 ($\rho = 0.77$, $p = 0.025$), as illustrated in Figure 5. In contrast, no statistically significant association was observed between Alpha activity and performance during the same activity ($\rho = 0.54$, $p = 0.167$). Given the small sample size and the exploratory nature of the analysis, these associations should be interpreted cautiously and require confirmation in larger studies.

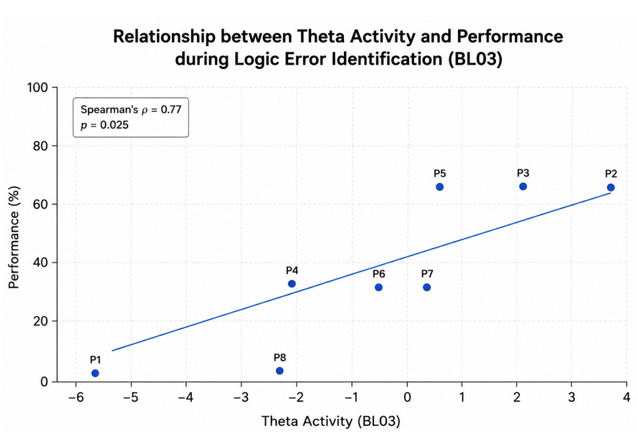

Figure 5: Relationship between Theta activity and participant performance during BL03. Each point represents one participant (P1–P8). A strong positive Spearman correlation was observed between Theta activity and task accuracy ($\rho = 0.77$, $p = 0.025$), suggesting an association between higher Theta activity and better task performance.

The positive correlation between Theta activity and performance suggests that participants exhibiting stronger Theta activation during BL03 tended to achieve higher accuracy. Since Theta activity is commonly associated with working memory engagement and cognitive processing, this pattern may reflect an association between Theta activity and successful fault identification.

Interestingly, Test Case Design (BL04) exhibited the strongest overall neural activation while maintaining relatively high performance levels. Conversely, Logic Error Identification (BL03) produced the lowest performance despite not exhibiting the highest activation levels across all EEG frequency bands. Together with the Alpha suppression observed in RQ2, these findings identify BL03 as a potentially relevant activity for further investigation of attentional and analytical demands.

Taken together, the results indicate that cognitive activation and task performance are related but not equivalent constructs. Activities producing stronger neural responses do not necessarily correspond to lower performance. Instead, the findings suggest that distinct software testing activities engage different cognitive mechanisms and that successful performance in fault identification may be associated with increased working-memory engagement.

RQ3 Summary. Participant accuracy varied descriptively across the four testing activities, with Logic Error Identification (BL03) showing the lowest observed accuracy. A strong positive association was observed between Theta activity and performance during BL03 ($\rho = 0.77$, $p = 0.025$). Overall, the observed patterns suggest that neural activation and task performance may be related, but the present data do not support causal or confirmatory conclusions.

5 Discussion

This study investigated neural activity and performance during four software testing activities: Test Code Comprehension (BL01), Syntax Error Identification (BL02), Logic Error Identification (BL03), and Test Case Design (BL04). The results provide preliminary descriptive evidence of variation in neural activity and performance across four software testing activities. Within this exploratory study, BL03 and BL04 showed comparatively stronger descriptive indicators of cognitive effort, although these patterns require confirmation in larger samples.

5.1 Cognitive Demands Across Testing Activities

The results suggest that software testing activities should not be treated as cognitively equivalent. While all activities required participants to analyze software artifacts and reason about program behavior, the observed EEG patterns provide preliminary indications that different testing activities may involve different cognitive demands.

Logic Error Identification (BL03) exhibited the largest reduction in Alpha activity relative to baseline and produced the lowest participant accuracy. Since Alpha suppression has been associated with increased attentional effort and cognitive workload [13], these findings suggest that identifying behavioral faults may require substantial analytical reasoning and sustained attention. The combination of reduced Alpha activity and poor performance further suggests that this activity showed the strongest descriptive indicators of cognitive demand in the present sample.

In contrast, Test Case Design (BL04) produced the largest increases in Theta and Delta activity while maintaining relatively high performance levels. This pattern suggests strong engagement of working-memory processes and mental planning without a corresponding decline in task effectiveness. This descriptive pattern is consistent with substantial cognitive engagement during BL04 despite relatively high observed performance, although this interpretation requires confirmation in larger samples.

These findings reinforce the notion that different testing activities rely on distinct cognitive resources and mental strategies. Consequently, studies investigating software testing behavior should consider the specific characteristics of each testing activity rather than treating testing as a homogeneous cognitive process.

5.2 Implications for Software Testing Practice

The observed patterns may have implications for future research on software testing practice and quality assurance processes.

First, Logic Error Identification showed the lowest observed accuracy and the largest reduction in Alpha activity in this sample, making it a potentially relevant target for further investigation and tool-support studies. This suggests that behavioral fault detection may benefit from additional methodological and technological support. For example, advanced debugging environments, fault localization techniques, and AI-assisted code analysis tools may help reduce the cognitive burden associated with understanding complex program behavior.

Second, the results indicate that cognitively demanding activities do not necessarily impair performance. Although Test Case Design produced the strongest neural activation patterns, participants maintained relatively high accuracy. This suggests that certain testing activities may require substantial cognitive effort while still allowing effective task execution when testers possess adequate knowledge and task understanding.

Finally, the observed relationship between Theta activity and performance during BL03 suggests that successful execution of challenging testing activities may depend on the ability to effectively allocate cognitive resources. This finding highlights the importance of training strategies that strengthen analytical reasoning, working-memory utilization, and systematic fault investigation skills.

5.3 Cognitive Hotspots for Tool Support

The observed patterns identify potential cognitive hotspots within software testing activities. In particular, Logic Error Identification and Test Case Design exhibited comparatively stronger descriptive indicators of cognitive effort.

From a software engineering perspective, these activities represent promising targets for intelligent tool support. Techniques such as AI-assisted debugging, automated fault localization, test case generation, and recommendation systems may help reduce cognitive effort while improving testing effectiveness.

Rather than uniformly supporting all testing activities, tool designers may achieve greater benefits by prioritizing activities that demand substantial attentional effort, reasoning, and working-memory engagement. The identification of cognitive hotspots may therefore contribute to more effective allocation of research and development efforts in software testing technologies.

5.4 Future Work

This study represents an exploratory investigation of neural activity during software testing and presents several opportunities for future research. First, the sample size should be expanded to improve statistical power and enable stronger inferential analyses. Second, future studies should investigate professional software testers and compare their neural activation patterns with those of less experienced participants.

Additional testing activities, including exploratory testing, regression testing, usability testing, and security testing, should also be examined to determine whether similar cognitive patterns emerge. Furthermore, combining EEG with complementary techniques such as eye tracking, physiological monitoring, or higher-resolution neuroimaging technologies may provide a more comprehensive understanding of the cognitive processes involved in software testing.

By further exploring the relationship between neural activity, cognitive load, and testing performance, future research may contribute to the development of testing methods and tools that better support software testers and improve software quality.

6 Threats to Validity

In this section, we describe the following threats to the validity of our study.

Construct Validity

A potential threat to construct validity concerns whether the experimental tasks adequately represent the software testing activities investigated in this study. To mitigate this threat, the experiment was designed around common testing activities routinely performed by professional software testers, including test code automation, syntax error identification, logic error identification, and test case design. These activities were selected because they capture distinct aspects of software testing practice and are representative of tasks frequently encountered in industrial settings.

Another potential threat relates to the measurement of cognitive load through EEG signals. Cognitive load is a complex construct that cannot be observed directly and must therefore be inferred from neurophysiological indicators. To reduce this threat, we relied on well-established EEG measures and analysis procedures that have been widely adopted in neuroscience and software engineering research. In addition, all participants performed the same tasks under controlled experimental conditions, reducing variability caused by environmental factors.

Task difficulty and time constraints may also influence participants' cognitive responses. To minimize this risk, the experimental tasks were carefully designed to be neither trivial nor excessively complex and were refined through a series of pilot studies. These pilot studies helped validate the experimental protocol, adjust task duration, and ensure that the activities could be completed within the constraints of the EEG experiment while remaining representative of realistic software testing scenarios [36].

Internal and External Validity

This study has several limitations that may affect its internal and external validity. First, the experiment was conducted in a controlled environment using relatively short testing tasks. While this design was necessary to minimize confounding factors and ensure reliable EEG data collection, it may not fully capture the complexity and dynamics of industrial software testing activities, where testers must continuously adapt to evolving requirements, larger code bases, and more complex testing scenarios [1]. Consequently, the cognitive demands observed in this study may differ from those experienced in real-world testing environments.

A second limitation concerns the sample size. The study involved eight professional software testers, which restricts statistical power and limits the generalizability of the findings. Consequently, the results should be interpreted as exploratory evidence regarding the cognitive demands of software testing activities rather than definitive conclusions.

The limited sample size may also have reduced the statistical power required to detect significant differences across activities. The largest observed *effect size* was found for Theta activity ($W = 0.206$), indicating a small observed effect that may warrant further investigation with larger samples.

Small sample sizes are common in EEG-based studies due to the substantial effort associated with participant recruitment, experiment preparation, signal acquisition, preprocessing, and analysis. Recruiting experienced software testing professionals presents an additional challenge, as the target population is considerably smaller and more specialized than the general population of software developers.

Although descriptive variation in neural activity was observed across testing activities, the Friedman tests did not reveal statistically significant differences between activities. This limitation is likely influenced by the small sample size and highlights the need for replication studies involving larger participant populations. Therefore, the findings should be interpreted as exploratory evidence until replicated with larger participant populations.

Finally, the experiment focused on four specific software testing activities. While these activities were selected because they represent common testing practices, other forms of testing such as exploratory testing, regression testing, usability testing, and security testing may involve different cognitive processes. Therefore, caution should be exercised when generalizing the results beyond the activities investigated in this study.

7 Conclusion and Future Work

Understanding the cognitive demands associated with software testing activities is important for both researchers and practitioners, as it may provide insights into the mental processes involved in testing effectiveness and software quality. Although previous studies have investigated cognitive aspects of software engineering activities, the perspective of software testers remains relatively underexplored. In this study, we conducted a controlled experiment using electroencephalography (EEG) to investigate neural activity and task performance during four software testing activities: Test Code Comprehension, Syntax Error Identification, Logic Error Identification, and Test Case Design.

Our results provide descriptive evidence of variation in neural activity and performance across the evaluated testing activities. Logic Error Identification (BL03) exhibited the largest reduction in Alpha activity and the lowest participant accuracy, suggesting greater attentional and analytical demands. Test Case Design (BL04) exhibited the highest Theta and Delta activity levels while maintaining relatively high performance, suggesting greater engagement of cognitive resources. However, the Friedman tests did not reveal statistically significant differences among activities in any of the analyzed EEG frequency bands. Therefore, the observed neural patterns should be interpreted as exploratory findings rather than evidence of statistically established differences in neural activation or cognitive demand.

The Theta band exhibited the largest observed effect size (Kendall's $W = 0.206$), indicating comparatively greater variation across testing activities than the other frequency bands. In addition, a strong positive association was observed between Theta activity and performance during Logic Error Identification ($\rho = 0.77$, $p = 0.025$). Given the exploratory nature of the analysis, this association should be interpreted as hypothesis-generating evidence rather than a confirmatory finding.

From a practical perspective, the observed patterns identify Logic Error Identification and Test Case Design as potentially relevant targets for further investigation of cognitive demands in software testing. These activities may provide promising contexts for studying whether methodological support, training strategies, automation techniques, and AI-assisted tools can help manage cognitive demands while preserving testing effectiveness.

Future work should involve larger samples to improve statistical power and assess the robustness of the observed patterns. Replication studies should also include a broader range of software testing activities, such as exploratory, regression, usability, and security testing, to examine whether similar cognitive demands emerge across different testing contexts. Finally, combining EEG with complementary measures, such as eye tracking and physiological monitoring, may provide a more comprehensive characterization of the cognitive processes involved in software testing.

Artifact Availability

To support reproducibility, all artifacts associated with this study are publicly available. The GitHub replication package[2] contains the experimental protocol, testing tasks, participant forms, the customized Prophet environment used during the experiment, MATLAB scripts for EEG preprocessing, Python scripts implementing the statistical analysis pipeline, and the statistical results reported in this paper.

The complete raw EEG dataset is available on Zenodo[3], including all EEG recordings collected during the experiment in European Data Format (EDF).

Acknowledgements

This study was partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, Finance

[2]Supplementary materials are available at: https://github.com/eeg-brain-study/experimental-study

[3]https://zenodo.org/records/21831259

Code 001. It was also partially supported by the National Institute of Science and Technology for Software Engineering Based on and for Artificial Intelligence (INES.IA), supported by CNPq under Grant No. 408817/2024-0.

References

[1] Pavlo Antonenko, Fred Paas, Roland Grabner, and Tamara Van Gog. 2010. Using electroencephalography to measure cognitive load. *Educational psychology review* 22 (2010), 425–438.

[2] Neil C. C. Brown, Pierre Weill-Tessier, Maksymilian Sekula, Alexandra-Lucia Costache, and Michael Kölling. 2022. Novice Use of the Java Programming Language. *ACM Trans. Comput. Educ.* 23, 1, Article 10 (dec 2022), 24 pages.

[3] Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha Crosby, James H Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. 2015. Eye movements in code reading: Relaxing the linear order. In *2015 ieee 23rd international conference on program comprehension*. IEEE, 255–265.

[4] Alessandra Calcagno, Stefania Coelli, Ricardo Couceiro, João Durães, Caterina Amendola, Ileana Pirovano, Rebecca Re, and Anna Maria Bianchi. 2020. EEG monitoring during software development. In *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*. IEEE, Palermo, Italy, 325–329.

[5] Camila Cosmo, Amin Zandvakili, Nicholas J. Petrosino, Thaise Graziele L. de O. Toutain, José Garcia Vivas Miranda, and Noah S. Philip. 2023. Examining the neural mechanisms of rTMS: a naturalistic pilot study of acute and serial effects in pharmacoresistant depression.

[6] Igor Crk, Timothy Kluthe, and Andreas Stefik. 2015. Understanding programming expertise: an empirical study of phasic brain wave changes. *ACM Transactions on Computer-Human Interaction (TOCHI)* 23, 1 (2015), 1–29.

[7] Madeline Endres, Zachary Karas, Xiaosu Hu, Ioulia Kovelman, and Westley Weimer. 2021. Relating Reading, Visualization, and Coding for New Programmers: A Neuroimaging Study. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, Madrid, ES, 600–612.

[8] Davide Fucci, Daniela Girardi, Nicole Novielli, Luigi Quaranta, and Filippo Lanubile. 2019. A replication study on code comprehension and expertise using lightweight biometric sensors. In *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*. IEEE, Montreal, QC, Canada, 311–322.

[9] William S. Greenwell, Elisabeth A. Strunk, and John C. Knight. 2004. Failure Analysis and the Safety-Case Lifecycle. In *Human Error, Safety and Systems Development*, Chris W. Johnson and Philippe Palanque (Eds.). Springer US, Boston, MA, 163–176.

[10] Fatih Gurcan, Gonca Gokce Menekse Dalveren, Nergiz Ercil Cagiltay, Dumitru Roman, and Ahmet Soylu. 2022. Evolution of Software Testing Strategies and Trends: Semantic Content Analysis of Software Research Corpus of the Last 40 Years. *IEEE Access* 10 (2022), 106093–106109.

[11] Thalía Harmony. 2013. The functional significance of theta oscillations in cognitive processing. *Frontiers in Integrative Neuroscience* 7 (2013), 58.

[12] Yoshiharu Ikutani and Hidetake Uwano. 2014. Brain activity measurement during program comprehension with NIRS. In *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, Las Vegas, NV, USA, 1–6.

[13] Wolfgang Klimesch. 1999. EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain research reviews* 29, 2-3 (1999), 169–195.

[14] Ryan Krueger, Yu Huang, Xinyu Liu, Tyler Santander, Westley Weimer, and Kevin Leach. 2020. Neurological divide: an fMRI study of prose and code writing. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering* (Seoul, South Korea) *(ICSE '20)*. Association for Computing Machinery, New York, NY, USA, 678–690.

[15] SeolHwa Lee, Andrew Matteson, Danial Hooshyar, SongHyun Kim, JaeBum Jung, GiChun Nam, and HeuiSeok Lim. 2016. Comparing programming language comprehension between novice and expert programmers using eeg analysis. In *2016 IEEE 16th international conference on bioinformatics and bioengineering (BIBE)*. IEEE, Taichung, Taiwan, 350–355.

[16] Yu-Tzu Lin, Yi-Zhi Liao, Xiao Hu, and Cheng-Chih Wu. 2021. Eeg activities during program comprehension: An exploration of cognition. *IEEE Access* 9 (2021), 120407–120421.

[17] Sergio Machado, Cláudio Portella, Julio Silva, Bruna Velasques, Patrícia Terra, Camila Ferreira-Vorkapic, Vernon Silva, Luis Miana, Luis Basile, Mauricio Cagy, Roberto Piedade, and Pedro Ribeiro. 2007. Changes in quantitative EEG absolute power during the task of catching an object in free fall. *Arquivos de neuro-psiquiatria* 65 (10 2007), 633–6.

[18] Christoph Mayr-Dorn, Michael Vierhauser, Stefan Bichler, Felix Keplinger, Jane Cleland-Huang, Alexander Egyed, and Thomas Mehofer. 2021. Supporting quality assurance with automated process-centric quality constraints checking. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, Madrid, ES, 1298–1310.

[19] Júlio Medeiros, Ricardo Couceiro, Gonçalo Duarte, João Durães, João Castelhano, Catarina Duarte, Miguel Castelo-Branco, Henrique Madeira, Paulo de Carvalho, and César Teixeira. 2021. Can EEG be adopted as a neuroscience reference for assessing software programmers' cognitive load? *Sensors* 21, 7 (2021), 2338.

[20] J. Medeiros, M. Simões, J. Castelhano, R. Abreu, R. Couceiro, J. Henriques, et al. 2024. EEG as a potential ground truth for the assessment of cognitive state in software development activities: A multimodal imaging study. *PLoS ONE* 19, 3 (2024), e0299108.

[21] Norman Peitek, Sven Apel, Chris Parnin, Andre Brechmann, and Janet Siegmund. 2021. Program comprehension and code complexity metrics: An fMRI study. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (Madrid, Spain). IEEE, Madrid, Spain, 524–536.

[22] Norman Peitek, Annabelle Bergum, Maurice Rekrut, Jonas Mucke, Matthias Nadig, Chris Parnin, Janet Siegmund, and Sven Apel. 2022. Correlates of programmer efficacy and their link to experience: a combined EEG and eye-tracking study. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Singapore, Singapore) *(ESEC/FSE 2022)*. Association for Computing Machinery, New York, NY, USA, 120–131.

[23] Norman Peitek, Janet Siegmund, Sven Apel, Christian Kästner, Chris Parnin, Anja Bethmann, Thomas Leich, Gunter Saake, and André Brechmann. 2018. A look into programmers' heads. *IEEE Transactions on Software Engineering* 46, 4 (2018), 442–462.

[24] Norman Peitek, Janet Siegmund, Chris Parnin, Sven Apel, Johannes C. Hofmeister, and André Brechmann. 2018. Simultaneous measurement of program comprehension with fMRI and eye tracking: a case study. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (Oulu, Finland) *(ESEM '18)*. Association for Computing Machinery, New York, NY, USA, Article 24, 10 pages.

[25] R. Pressman. 2014. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education. https://books.google.com.br/books?id=dXlzCgAAQBAJ

[26] Patrick Rempel, Patrick Mäder, Tobias Kuschke, and Jane Cleland-Huang. 2014. Mind the gap: assessing the conformance of software traceability to relevant guidelines. In *Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE 2014)*. Association for Computing Machinery, New York, NY, USA, 943–954.

[27] Saeid Sanei and J.A. Chambers. 2007. *Introduction to EEG*. John Wiley & Sons, Ltd, Chapter 1, 1–34.

[28] Samuel Sanford Shapiro and Martin B Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52, 3-4 (1965), 591–611.

[29] Siegel Sidney. 1957. Nonparametric statistics for the behavioral sciences. *The Journal of Nervous and Mental Disease* 125, 3 (1957), 497.

[30] Janet Siegmund, André Brechmann, Sven Apel, Christian Kästner, Jörg Liebig, Thomas Leich, and Gunter Saake. 2012. Toward measuring program comprehension with functional magnetic resonance imaging. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering* (Cary, North Carolina) *(FSE '12)*. Association for Computing Machinery, New York, NY, USA, Article 24, 4 pages.

[31] Janet Siegmund, Christian Kästner, Sven Apel, Chris Parnin, Anja Bethmann, Thomas Leich, Gunter Saake, and André Brechmann. 2014. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE 2014)*. Association for Computing Machinery, New York, NY, USA, 378–389.

[32] Janet Siegmund, Christian Kästner, Sven Apel, André Brechmann, and Gunter Saake. 2013. Experience from measuring program comprehension - toward a general framework.

[33] Janet Siegmund, Norman Peitek, Chris Parnin, Sven Apel, Johannes Hofmeister, Christian Kästner, Andrew Begel, Anja Bethmann, and André Brechmann. 2017. Measuring neural efficiency of program comprehension. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (Paderborn, Germany) *(ESEC/FSE 2017)*. Association for Computing Machinery, New York, NY, USA, 140–150.

[34] Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, and Qing Wang. 2024. Software Testing With Large Language Models: Survey, Landscape, and Vision. *IEEE Transactions on Software Engineering* 50, 4 (2024), 911–936.

[35] Barbara Weber, Thomas Fischer, and René Riedl. 2021. Brain and autonomic nervous system activity measurement in software engineering: A systematic literature review. *Journal of Systems and Software* 178 (2021), 110946.

[36] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[37] Marvin Wyrich, Justus Bogner, and Stefan Wagner. 2023. 40 years of designing code comprehension experiments: A systematic mapping study. *Comput. Surveys* 56, 4 (2023), 1–42.

[38] Martin K-C Yeh, Dan Gopstein, Yu Yan, and Yanyan Zhuang. 2017. Detecting and comparing brain activity in short program comprehension using EEG. In *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, Indianapolis, USA, 1–5.