TIDEs

Technostress-Aware Integrated Development Environments

by

Sasan Radan

School of Computing

BIRMINGHAM CITY UNIVERSITY

Supervised by *Dr Sara Hassan*

A thesis submitted for the degree of

Doctor of Philosophy

1st March 2026

Preface

This thesis represents the culmination of several years of research into the intersection of mental wellbeing, technostress, and software design within programming environments. The work originated from a simple but important question: "*Can the tools we use to code be designed to care for the people who use them?*". Throughout my academic and professional journey, I have observed how modern Integrated Development Environments (IDEs), despite their sophistication can often contribute to cognitive overload particularly for users learning to program. According to Higher Education Statistics Agency (HESA) data for the 2019/20 academic year, 7.7% of Computing undergraduates did not continue their studies after the first year, compared with 5.3% across all subjects [1]. Although student attrition is recognised as a multifactorial issue, the comparatively higher non-continuation rate within Computing provides contextual evidence that students in this discipline may experience distinctive academic and cognitive pressures. Within this broader context, programming-environment challenges and technostress-related factors may represent one possible contributing influence among several others affecting student engagement, performance, and continuation. Taken together, personal experience and sector-level data highlight that the challenges associated with programming environments are not anecdotal but form part of a wider pattern affecting computing students at scale. Despite increasing recognition of technostress in broader technological contexts, technostress within programming environments remains insufficiently examined as a measurable and design-addressable phenomenon. Technostress refers to the psychological strain and negative cognitive or emotional responses experienced as a result of interacting with information technologies, particularly when technological demands exceed an individual's coping resources. It is commonly conceptualised through five primary stressors: techno-overload, where technology forces users to work faster or process excessive information; techno-complexity, where systems are perceived as difficult to understand or use; techno-insecurity, where users feel threatened about their competence or fear being replaced due to insufficient technical skills; techno-uncertainty, where constant updates or unclear system behaviour create instability and unpredictability; and techno-invasion, where technology intrudes into personal space, disrupts workflow, or creates a sense of constant connectivity. Developers, as implementers of change, often lack structured, feedback-driven methodologies that explicitly increase recognition of technostress in broader technological contexts, technostress within programming environments remains insufficiently examined as a measurable and design-addressable phenomenon. Consequently, a disconnect persists between users lived experiences of technostress and the structured mechanisms required to systematically translate those experiences into actionable design refinement within development environments. This observation formed the motivation for developing the Technostress-aware Integrated Development Environments (TIDEs) project: a research endeavour exploring how software design can be reimagined to support mental wellbeing, reduce technostress, and enhance user experience within programming contexts. The study unfolds across several core contributions. First, it establishes a taxonomy exploring the intersection between technostress, mental health, and assistive technology research to systematically synthesise and categorise it. Second, the taxonomy helps derive a set of derived metrics, performance, productivity, and satisfaction to measure the effects of design decisions on user's technostress levels. Third, it operationalises these metrics an Iterative IDE Enhancement Framework, which facilitates integrating user feedback and biometric data to iteratively refine IDEs in ways that respond to emotional and cognitive needs with a particular focus on technostress. Within the scope of this work, the primary focus is on Integrated Development Environments, with designers and developers of these systems positioned as the primary users of the proposed contributions. Students are considered secondary users and beneficiaries, as the framework and dataset are intended to inform the design decisions that shape their programming experiences. Together, these contributions seek to move the conversation in computing beyond efficiency and functionality, towards a more human-centred vision of technology, one that recognises wellbeing as a critical component of effective design. In addition to the framework, the development and validation of the TIDE dataset provide an empirical foundation for this shift, offering structured emotional and interaction data that support evidence-based technostress-aware design and future affective computing research. Evaluating the framework, demographic factors such as gender, educational level, programming experience, and ethnicity significantly influenced user perceptions of the modified IDE. Additionally, the tailored modifications led to measurable improvements in performance, productivity, and satisfaction, reductions in technostress, with clear mappings between interface changes, user outcomes, and specific technostress triggers. These results offered a practical insight for designing more supportive and technostress-focused development environments that align with diverse user needs. As for the biometric data, key findings showed that positive emotional markers such as joy and engagement correlate with higher task success, while negative affect is linked to underperformance. Additionally, emotional trajectories improved under the modified IDE, suggesting that design interventions can reduce technostress. With TIDE dataset offering a unique foundation for researchers aiming to understand the role of technostress in digital environments and to build more technostress-aware IDEs. The evaluation phase, involving both professional developers and MSc User Experience Design participants, support that the proposed technostress-aware framework and the accompanying TIDE biometric dataset were perceived as credible, relevant, and complementary resources. The combined findings from Chapter 5 indicate that, when evaluated by their primary users and implementers, both the framework and the TIDE dataset were regarded as methodologically sound, practically applicable, and effective in supporting an evidence-based approach to identifying and mitigating technostress in programming environments.

Acknowledgements

I would like to express my sincere gratitude to all the participants who contributed to this research. To the undergraduate students, thank you for taking part in the experimental study and for your time, commitment, and openness throughout the data collection process. To the postgraduate students and professional developers, I am deeply grateful for your valuable insights and constructive feedback during the framework and dataset evaluation stages. Your professional experience and academic perspectives provided critical depth and relevance to the study, helping shape its real-world applicability and future direction. This research would not have been possible without the engagement, curiosity, and collaboration of every participant. Your contributions have not only informed the development of technostress-aware IDEs but have also played a meaningful role in advancing understanding of technostress and mental wellbeing in computing education and practice. Furthermore, I would also like to extend my sincere thanks to Dr Samer Bamansoor and Dr Abdulrahman Alsewari for generously sharing this research with their MSc cohorts. Their support in encouraging participation was invaluable, helping to secure rich, high-quality feedback that strengthened both the evaluation of the framework and the TIDE dataset.

I would like to express my deepest gratitude to the most important figure throughout this journey, my Director of Studies, Dr Sara Hassan. From day one, your unwavering guidance, encouragement, and invaluable support have been instrumental in shaping both this research and my academic development. I feel incredibly fortunate to have had a supervisor who genuinely cares, continuously motivates me, and shares knowledge with such passion and generosity.

Your mentorship has not only shaped this PhD but has also played a pivotal role in developing my confidence, curiosity, and understanding of research. Everything I know about research today is largely thanks to your guidance. Throughout this journey, you consistently encouraged me to see possibilities where I saw obstacles and helped me maintain perspective during moments when I doubted myself. *When I found myself in the dark, uncertain of the path ahead, you were there with a lamp, not to fix me or solve every problem, but simply to sit beside me until I could see the way forward again.*

Thank you for your kindness, patience, insight, and for being there every step of the way, every week since day one. Your belief in me often came at times when I struggled to believe in myself. Without your support, encouragement, and unwavering faith, this PhD would not have been possible.

~ to my family ~

While this degree may carry my name, the journey behind it carries your sacrifices.

Thank you, Maman and Baba, for everything you have done for me. Throughout my life, I have witnessed the challenges you have faced, yet I have never seen either of you give up. Your resilience, determination, and unwavering commitment has been a constant source of inspiration.

The clock never stops ticking, and pursuing this dream often came at the cost of precious time. Yet, in many ways, this dream was always for you. I can only hope that God grants me enough time and opportunity to show you how deeply grateful I am for all that you have done and continue to do.

For every sacrifice, every prayer, every struggle, and every moment you stood beside me, thank you. Your belief in me carried me through the most difficult moments, and your love gave me the strength to keep moving forward when the path seemed uncertain.

No words can truly express the depth of my gratitude. This achievement is as much yours as it is mine.

With love,
Your Son.

Declaration

I, Sasan Radan, solemnly affirm that the research work presented in this thesis, titled "*Technostress-aware Integrated Development Environments (TIDEs)*", is entirely my own work, except where explicit acknowledgements are made. This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) at Birmingham City University. All sources of information, ideas, and data have been properly cited and referenced in accordance with established academic conventions. The research was conducted in full compliance with the ethical standards and guidelines of Birmingham City University. I confirm that this thesis has not been submitted, either in whole or in part, for any other academic award or qualification. However, I note that a portion of this research has been accepted for publication on behalf of Birmingham City University, in alignment with the University's commitment to promoting the dissemination of knowledge for the benefit of the academic community. All contributions made by others have been duly acknowledged, and all data, analyses, and conclusions presented within this work are authentic and original. I am fully aware of the serious implications of plagiarism and academic dishonesty and affirm that this thesis represents my own independent academic effort. I make this declaration with complete understanding of its significance and with full awareness of the responsibilities and ethical obligations it entails.

Publications

During the course of this research, several scholarly outputs have been produced. One paper (Taxonomy) has been formally published [P1], while two additional papers have been accepted for publication and are currently undergoing final revisions prior to release [P2, P4]. In parallel, two research datasets have been systematically curated to support transparency, reproducibility, and extended analysis. Dataset [D1] accompanies paper [P3], and dataset [D2] has been prepared in support of paper [P4], ensuring alignment between empirical outputs and their corresponding publications. Additionally, a supplementary paper has been published that aligns conceptually with the broader research trajectory of this thesis, although it is not formally included as a core component of the thesis submission [P2]. Collectively, these outputs support the progressive dissemination of the research findings, the integration of empirical artefacts, and the contribution of both methodological and applied insights to the academic community.

Papers

Published

[P1] Radan, Sasan and Hassan, Sara and Wilson, Andrew and Basurra, Shadi. (2023). A Review of Mental Health Issues Prevalent in Science, Technology, Engineering, and Mathematics (STEM) Subjects. Handbook of research on innovative frameworks and inclusive models for online learning, 1-27, https://doi.org/10.4018/978-1-6684-9072-3.ch001.

Accepted

[P2] Hassan, Sara and Ali, Nour and Radan, Sasan and Wilson Andrew and Galvin, John and Ahmed Anis (2025) A Meta-Model for Technostress-Aware Software Architecture Design. Springer Nature, Journal of Personal & Ubiquitous Computing

[P4] Radan, Sasan and Wilson, Andrew and Basurra, Shadi and Hassan, Sara. (2026). TIDE: Technostress in IDEs - A Dataset for Exploring Technostress During Programming Tasks in Integrated Development Environments Enhancement. International Journal of Human–Computer Interaction.

To be Revised

[P3] Radan, Sasan and Hassan, Sara and Wilson, Andrew and Basurra, Shadi. (2026). TIDE: Technostress in IDEs - A Framework for Technostress-Centric IDE Enhancement. International Journal of Human–Computer Interaction.

Datasets

Published

[D1] Radan, Sasan and Hassan, Sara and Wilson, Andrew and Basurra, Shadi (2025) TIDE: Technostress in IDEs Dataset - Quantitative and Qualitative Data. Available at https://www.open-access.bcu.ac.uk/16622/ .

[D2] Radan, Sasan and Hassan, Sara and Wilson, Andrew and Basurra, Shadi (2025) TIDE: Technostress in IDEs Dataset - Biometric Data. Available at https://www.open-access.bcu.ac.uk/16623/ .

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

Abbreviation	*Meaning*	*Page*
TIDE	Technostress-aware Integrated Development Environment	15
IDE	Integrated Development Environment	15
HCI	Human Computer Interaction	15
AC	Affective Computing	15
STEM	Science Technology Engineering Mathematics	16
HE	Higher Education	16
AT	Assistive Technology	18
UI	User Interface	18
ICT	Information Communication Technology	25
WHO	World Health Organization	26
PwD	People with Disability	27
PnwD	People Not with Disability	27
SUS	System Usability Scale	56
SS	SUS Score	56
TAM	Technology Acceptance Model	57
PU	Perceived Usefulness	57
PEOU	Perceived Ease of Use	57
SMI	System Modification Impact	58
TA	Task Assessment	58
M	Mean	60
SD	Standard Deviation	60
SEM	Standard Error of the Mean	60
CA	Cronbach's Alpha	60
HIR	High Internal Reliability	60
LIR	Low Internal Reliability	60
UEI	User Experience Insight	62
TRQ	Task Requirements	64
TTC	Task Test Case	64
TCT	Task Completion Time	64
TC	Task Completion	64
LLM	Large Language Models	65
RA	Recommended Action	65
BCU	Birmingham City University	66
SoC	School of Computing	66
SDLC	Software Development Life Cycle	68
SFT	Supervised Fine-Tuned	72
AOI	Areas of Interest	74
EEG	Electroencephalogram	117
ECG	Electrocardiogram	117
EMG	Electromyography	117
GSR	Galvanic Skin Response	117
FE	Facial Expression	117
FEA	Facial Expression Analysis	118

Chapter *One*

1 Introduction

This chapter introduces the research context and motivation underpinning the Technostress-aware Integrated Development Environment (TIDEs) project. It defines the overarching research problem, outlines the research aim and questions, and establishes the rationale for developing a technostress-aware Integrated Development Environments (IDEs). The chapter also presents the key theoretical, methodological, and empirical contributions of the study. Additionally, it discusses the broader motivation and impact of the research, supported by a scenario that illustrates its real-world relevance. The chapter concludes with an overview of its structure to guide the reader through the subsequent sections.

1.1 Background

The design of software systems has undergone a significant transformation over the last half-century. Early computing environments of the 1960s and 1970s were heavily system-centred, built around machine constraints, limited memory, and command-line interaction. Users were expected to adapt to the system rather than the system adapting to them. Software interfaces were abrupt, cryptic, and optimised primarily for efficiency and correctness rather than cognitive or emotional experience. This period marked what many scholars describe as the engineering-driven era of software design, where human factors were seldom considered beyond basic usability [2].

The 1980s introduced a fundamental paradigm shift through the emergence of Human Computer Interaction (HCI) as a scientific discipline. HCI reframed software not merely as a computational artefact but as a medium of human activity shaped by perception, cognition, and behaviour. Pioneering contributions such as Shneiderman's principles of direct manipulation [3] and Norman's work on human-centred design [4] emphasised that interfaces must support mental models, minimise cognitive load, and facilitate error recovery. This era saw the commercial rise of graphical user interfaces, usability engineering, and systematic user-centred design methodologies. The role of the human, rather than the machine, became the focal point of interface development.

As software integrated more deeply into everyday life, HCI scholarship expanded from purely cognitive concerns toward emotional, experiential, and sociocultural dimensions of technology use. The late 1990s introduced a pivotal development, affective computing, formalised in Picard's landmark monograph, which argued that computers must be capable of recognising and responding to human emotions to support genuinely effective interaction. This positioned emotion as a functional component of interface design, with implications for learning, decision-making, and problem-solving [5]. Subsequent empirical studies indicate that emotional states are closely tied to task performance, cognitive strain, and sustained attention, core challenges in programming environments. Research by [6] and [7] further advanced the field by establishing multimodal affect detection techniques, framing emotion as a measurable and computationally actionable construct.

Parallel to these developments, research in technostress [8] began to highlight the psychological costs of continuous digital engagement. While early HCI sought to make systems more usable, technostress research revealed that usability alone was insufficient if technology simultaneously imposed cognitive overload, emotional fatigue, or a sense of loss of control. This realisation inspired a convergence between Affective Computing (AC) and technostress mitigation research.

From this historical arc, from system-driven design to human-centred methods, to affective and technostress-aware computing, the need for technostress-sensitive, emotion-informed IDEs emerges as a natural and necessary progression. The TIDEs project positions itself at the forefront of this evolution. Building upon decades of HCI, AC, and technostress research, it aims to reimagine IDE design through the lens of emotional intelligence and cognitive ergonomics. By combining biometric insight, user feedback, and iterative refinement, the TIDEs approach aspires to establish a systematic, evidence-based framework for creating IDEs that not only enhance performance and productivity but also improve satisfaction for sustained mental wellbeing and reduce technostress among end-users. Techno-stressors (Table 1.1) are the triggers, events, and circumstances that cause technostress [9], all of which are moderately correlated with each another and often occur simultaneously [10] [8].

Table 1.1: Technostress Terms

Stressor	***Definition***
Techno-Overload	When the technology usage results in high workload, resulting in the user feeling forced to work faster and longer.
Techno-Complexity	The difficulty of understanding and using technology due to its complexity.
Techno-Invasion	The feeling that technology is constantly intruding into one's personal or work life.
Techno-Insecurity	Anxiety about losing job relevance due to rapid technological advancements.
Techno-Uncertainty	Stress caused by frequent changes in technology that make it difficult to adapt to.

Technostress is conceptually distinct from usability and technology acceptance. While usability focuses on the effectiveness, efficiency, and satisfaction associated with system use, and technology acceptance examines users' intentions and willingness to adopt a technology, technostress concerns the psychological strain, cognitive burden, and adverse emotional responses that may arise from interacting with technology. Consequently, improvements in usability or acceptance do not necessarily imply reductions in technostress, although they may influence factors associated with it. Within this thesis, technostress is examined through a combination of self-reported, behavioural, and biometric measures, rather than being inferred solely from usability or acceptance outcomes.

1.2 Motivation

While the importance of mental wellbeing and technostress in educational technology is increasingly recognised, the true impact of these issues becomes evident when examined through the lived experiences of students.

The first scenario illustrates a realistic situation that many students face when completing programming tasks in higher education. It demonstrates how technostress can arise from common challenges. While individually these issues may seem minor, together they can accumulate to create significant barriers to learning and wellbeing. The second scenario focuses on the developer's perspective as the implementer of solutions through the proposed TIDEs framework.

1.2.1 Motivation Scenario

Consider a scenario where a first-year undergraduate Science Technology Engineering and Mathematics (STEM) student is studying online due to COVID, personal, or work-related circumstances. These hours are all spent in a seated position in front of a laptop screen where the laptop is used both for personal and educational purposes. On any given weekday, about an hour is initially spent reading a programming task on the screen which was asynchronously provided by the teacher. The task instructions include step by-step instructions and screenshots of the outputs and inputs of each task. The screenshots are in a different version of software from the one the student is using. The student therefore spends about 2 hours researching how to install and set up an IDE that is compatible with their laptop and how to translate the instructions from the given task to the IDE version at hand. This initial task already exposes the student to techno-overload.

The following 2-3 hours are then spent to complete the given programming task after setting up the required IDE. The written code now includes errors. The error is explained using technical logs unreadable to the student. This common situation now leads to techno-complexity. Finally, the student resorts to trial and error making insignificant changes every time in hope for the program to produce the required output; eventually the output is as required. The student however does not understand why. This now leads to techno-reliability issues since the IDE was not consistent as far as the student is concerned. The above common narrative can happen fully or partially on any given day for a STEM student or in fact for any student. It is scenarios such as the above that motivate us to propose TIDEs in order to aid in reducing some of the technostress factors that STEM Higher Education (HE) students can experience when using "traditional" IDEs.

Now consider the perspective of a developer, the individual responsible for designing and maintaining these learning environments. In practice, developers face several challenges when attempting to improve students' experiences in relation to technostress. First, there is often limited clarity regarding what aspects of user behaviour or experience should be measured to capture technostress meaningfully. Second, even when issues are identified, developers lack clear guidance on what IDE design adjustments can be made with a specific focus on technostress, or how such adjustments should be evaluated. Finally, there is no established benchmark that enables developers to interpret users' real-time emotional and behavioural responses to IDE interactions in a structured and evidence-driven manner.

1.2.2 Problem Statement

Technostress emerges through routine interactions with programming and development tools that are essential for both education and professional practice. For students, these experiences can disrupt learning processes, increase cognitive strain, and undermine overall wellbeing. Over time, such challenges may extend beyond the classroom and influence how individuals engage with technology in professional contexts. Despite the central role of IDEs, technostress within these environments remains insufficiently examined as a measurable and design-addressable phenomenon. Furthermore, today's students are tomorrow's developers. The way students experience and cope with technostress during their studies directly shapes how they approach software design in their future careers. If educational tools induce unnecessary cognitive overload or poorly managed interaction complexity, these patterns risk being normalised and unintentionally reproduced in the systems developers later create. Technostress therefore represents not only a user-experience issue, but a systemic design concern with long-term implications for software engineering practice.

At the same time, developers who act as implementers of change, lack structured, feedback-driven methodologies that explicitly integrate technostress considerations into IDE design. While performance and usability are commonly evaluated, emotional sustainability and technostress mitigation are rarely embedded within iterative development cycles in a systematic manner. This creates a disconnect between users lived experiences of stress and the processes used to refine development environments. Together, these perspectives highlight a critical gap: the absence of an empirically grounded, iterative approach capable of capturing technostress data and translating it into actionable design refinement.

1.3 Scope

In this study, while the primary motivation centres on addressing technostress in educational settings, particularly for STEM students, the broader landscape of software development environments is also considered. Chapter Two includes both educational and workplace contexts to provide a comprehensive taxonomy of factors influencing technostress. This dual focus is important because it recognises that many of the usability, cognitive, and emotional challenges associated with IDE usage extend across both academic and professional software-development environments.

The intended implementers and primary users of the proposed framework are IDE designers, software developers, researchers, and tool engineers responsible for designing or modifying development environments. The framework is therefore positioned as a design-support and evaluation mechanism intended to guide the creation of more adaptive, user-centred, and technostress-aware IDE environments. Chapter Five further supports this perspective through the developer-focused evaluation, where feedback is collected regarding the practicality, usability, and implementation relevance of the framework from a development and design standpoint.

At the same time, computing students serve as the empirical participant group and represent the immediate end-user beneficiaries of the research. Students were selected because they engage intensively with IDEs in cognitively demanding educational settings and provide a highly relevant population for investigating IDE-induced technostress, emotional responses, and interaction behaviours. Consequently, the biometric dataset, experimental findings, and framework validation are grounded in real user evidence derived from student programming activities.

Accordingly, while the taxonomy in Chapter Two incorporates workplace literature to establish broader conceptual foundations, the later chapters narrow the experimental and analytical focus towards educational programming environments. This ensures that the resulting framework, dataset, and design recommendations remain directly applicable to student-centred educational contexts while still providing transferable insights for future IDE development and professional software-engineering applications.

1.4 Research Aim and Objectives

The aim of the TIDEs project is to produce a set of inter-related contributions for designing IDE software in a way that is aware of and supports users' mental wellbeing and mitigates technostress. The project uses IDEs as a case study to explore how software design decisions can influence user experience and technostress.

The primary beneficiaries of this research are STEM higher education students, who frequently engage with IDEs as part of their coursework and assessments. The primary users of the resulting contributions are software designers and developers, particularly those involved in creating or enhancing IDEs for STEM learning environments.

1.5 Research Questions

To achieve this aim, this study advances research through a combination of theoretical, methodological, and empirical contributions, guided by the following objectives:

(Obj 1) To develop a taxonomy and set of derived metrics that conceptualise the relationship between technostress research and related Mental Health and Assistive Technology (AT) research fields.

(Obj 2) To design and implement an Iterative IDE Enhancement Framework that facilitates integrating user feedback and biometric technostress indicators to guide iterative IDE improvements.

(Obj 3) To construct a dataset, capturing biometric data during cyclical programming tasks, enabling objective assessment of user experience.

(Obj 4) To evaluate the proposed framework and dataset through expert perspectives.

To achieve these objectives, the study addresses the following overarching research questions:

(RQ 1) How can technostress and its influencing factors be systematically studied and categorised?

(RQ 2) How can established iterative design and user-centred evaluation practices be adapted into a feedback-driven, technostress-aware framework that guides the development of IDEs supporting technostress reduction and mental wellbeing?

(RQ 3) How can a reusable emotional dataset, captured during IDE-based programming tasks, be constructed and analysed to inform technostress-centric User Interface (UI) design and decision making?

(RQ 4) How can the proposed framework and dataset be evaluated in an end-to-end manner through expert assessment?

These four overarching research questions form the foundation of this thesis and will be revisited in Chapter Six. The chapters in between focus on addressing specific aspects of each question through smaller, targeted research questions aligned with each stage of the investigation. These chapter-level research questions are not intended to operate independently; rather, they serve to operationalise the overarching thesis research questions by breaking them into manageable and evaluable components. In particular, Chapters Two, Three, and Four focus respectively on the taxonomy development, the framework design, and the dataset creation, with each chapter addressing a distinct subset of objectives that collectively contribute towards answering the overarching research questions presented above.

1.6 Contributions

The contributions of the research provide a structured foundation for understanding key factors influencing mental health, a practical approach for integrating technostress mitigation into IDEs, and empirical evidence to validate the proposed framework.

1.6.1 Theoretical Contribution – Taxonomy and Metrics

Presenting a comprehensive novel taxonomy that consolidates and categorises the key factors affecting mental health within STEM, particularly in relation to technostress. This taxonomy enabled the identification of clear gaps in existing research, which subsequently became the focal points of the study and shaped its overall direction. Building upon the taxonomy, a set of derived metrics from the literature that directly informed the next contribution.

1.6.2 Methodological Contribution – Procedural Framework

The design of a structured framework aimed at addressing the identified research gaps. This novel procedural framework provides a methodological basis for enhancing IDEs with a focus on technostress mitigation. It provides a structured user-feedback-driven approach to iteratively design and evaluate technostress-aware IDE designs.

1.6.3 Empirical Contribution – Biometric Data

Through experimentation, novel real-time biometric data was collected, capturing emotional responses during IDE usage. These data provide empirical support for the framework's outcomes, reinforcing the methodological findings and offering an evidence based additional layer of depth by delivering direct, real-time emotional insights into the impact of technostress-aware design.

1.7 Impact and Significance

The impact of this research can be understood across three key dimensions: students as beneficiaries, developers as implementers, and broader applications beyond IDEs.

1.7.1 Impact on Students

For STEM students, particularly those studying computing, this research provides direct and meaningful benefits. By identifying how technostress manifests within IDEs, the study highlights the psychological and cognitive strain students experience when working with complex software tools. Through its contributions, the research demonstrates how IDEs can be designed to reduce technostress and improve outcomes. This technostress-centred approach supports improved academic performance and productivity and satisfaction. In the long term, it contributes to better mental health, self-efficacy, and degree attainment among students. According to Higher Education Statistics Agency (HESA) data for the 2019/20 academic year, 7.7% of Computing undergraduates did not continue their studies after the first year, compared with 5.3% across all subjects [1]. This makes Computing a discipline with one of the highest above average non-continuation rates in UK HE, reflecting the pressures that accompany students. Stats by [11] indicate that second-year undergraduate computing students in HE experienced higher-than-average non-continuation and low engagement, with reports linking this to emotional strain and tool related challenges. This evidence underlines the necessity of designing programming environments that actively support mental wellbeing, not just functional efficiency.

1.7.2 Impact on Implementers

For software developers and system designers, the TIDEs research offers a clear set of practical contributions. The derived metrics from the taxonomy and iterative framework provide implementable guidance for developing technostress-aware IDEs. The research encourages a shift in development culture where usability and outcomes are balanced with wellbeing considerations. By embedding these principles into the software design process, developers can create IDEs that promote a more positive, mindful user experience. This human-centred approach has the potential to redefine design standards within educational and professional software ecosystems.

1.7.3 Broader Applications

Beyond the immediate context of IDEs, the principles and findings of this research hold broader relevance across technology enhanced environments. In today's highly digital workplaces, professionals across industries face similar technostress triggers such as overload, complexity, and constant connectivity, which can undermine wellbeing and job satisfaction. The insights gained through the TIDEs project can therefore inform the design of technostress-aware systems in diverse domains (e.g. healthcare digital technologies, financial software, creative graphical design tools). By integrating technostress metrics and adaptive feedback mechanisms, organisations can cultivate healthier digital ecosystems that prioritise both outcomes and psychological resilience.

1.8 Structure

The structure of this thesis follows a sequential yet iterative process, as illustrated in Figure 1.1. It progresses through distinct but interlinked stages, represented by both solid and dotted boxes. The solid boxes denote the core contribution chapters that form the main body of the research, while the dotted boxes represent supporting components such as the overarching research problem and conclusion that connect and reinforce the main stages. The dotted line represents the iterative process of data accumulation across cycles, while the dashed line represents the supporting components that frame and contextualise the core contributions. Together, these chapters represent an interconnected process, from conceptualisation to validation, demonstrating how theory, data, and user experience come together to advance the development of technostress-aware and technostress-mitigating programming environments.

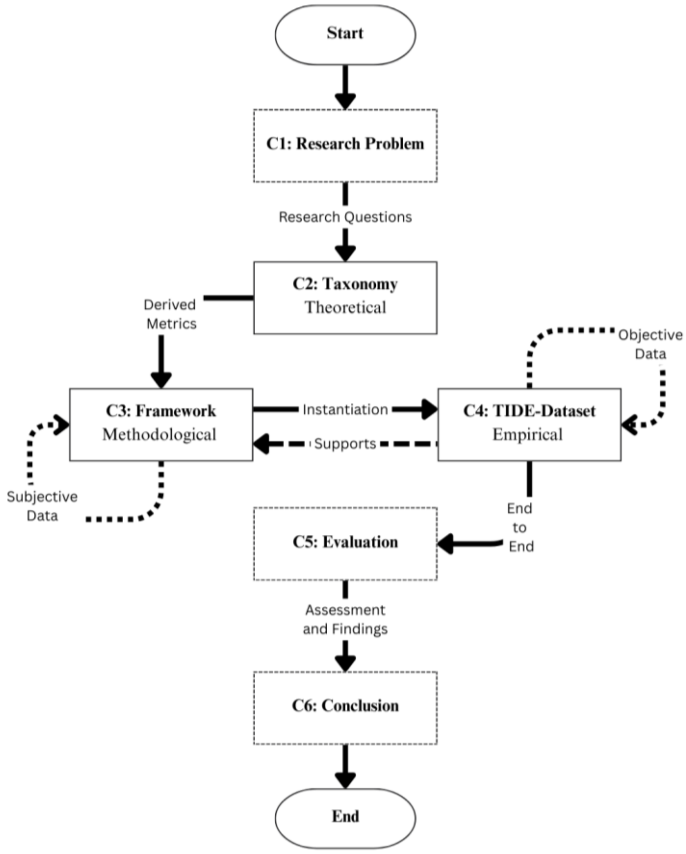


Figure 1.1: Thesis Structure

Chapter 1 introduces the research context, motivation, and the key challenges addressed in the study. It articulates the overarching research problem and defines the research questions that guide the investigation. The discussion outlines the rationale for exploring emotional and cognitive dimensions in programming environments and establishes the significance of developing a technostress-aware IDEs and an emotion-informed framework.

The then leads to Chapter 2 which expands on the research problem by systematically identifying and classifying the key themes central to this study. It develops a taxonomy that maps the interrelationships between mental health, technostress, and AT. The taxonomy serves as both a conceptual and theoretical foundation for the research, guiding the development of derived metrics, that are used throughout later chapters.

Chapter 3 details the design and architecture of the proposed framework. The framework enables iterative collection of both subjective data (e.g., self-reports and feedback) and objective biometric data (e.g., facial emotion indicators). This framework operationalises the derived metrics from the taxonomy. The illustrative usage of the framework in this chapter focuses on using subjective data to suggest iterative refinements to IDE designs. The framework also supports the development of a feedback-based system capable of linking emotional states to programming phases and IDE features.

Chapter 4 introduces the TIDE dataset, a novel time-series biometric dataset collected during programming tasks, using both baseline and modified IDEs. It documents the data collection methodology, including participant recruitment, instrumentation, and experimental protocol. The dataset contains synchronised streams of emotional and interaction data, serving as both an input to and validation source for the framework, from which representative emotional markers were derived to support subsequent analysis and design decisions. It represents the objective foundation of the study's analysis.

Thereafter, Chapter 5 evaluates the proposed framework and the TIDE dataset through survey-based feedback collected from two expert groups: developers from industry and MSc Computing students enrolled on the UX/UI Design degree. The evaluation focuses on assessing the perceived usefulness, novelty, and future value of the technostress-aware, feedback-driven IDE enhancement framework. In addition, participants provided insights into the TIDE dataset, evaluating its novelty, research utility, and potential for replication within academic and applied contexts. The findings from this evaluation offer a critical reflection on the framework's conceptual validity and practical relevance, informing the concluding chapter that discusses the overall contributions, implications, and directions for future research.

Finally, chapter 6 synthesises the study's outcomes, summarising key findings, theoretical implications, and practical contributions. It discusses how the framework and dataset advance understanding in technostress-aware and affective software engineering of IDEs. Limitations of the study are acknowledged, and directions for future research are proposed, emphasising the scalability and interdisciplinary applicability of the findings.

Chapter *Two*

2 Taxonomy

2.1 Introduction

This chapter brings together three central themes: mental health, technostress, and AT within educational and workplace contexts. Synthesising users' interaction with technology through a mental health lens brings technostress rightly to the forefront of designing technology mediated environments. Synthesising advancements in AT, particularly related to mental health establishes an understanding of gaps in that field paving the way for addressing them. Such gaps are of particular significance to users in STEM fields, where the use of technology for long periods of time is a core requirement in educational and workplace contexts.

The key motivation behind this comprehensive survey is to systematically review the existing literature on mental health within STEM education and workplace settings (to be addressed as part of RQ2). This chapter explores the definition and scope of mental health, technostress, and AT as they relate to STEM students and IT professionals. It aims to collate through a systematic literature review/mapping study the most significant factors affecting mental health, examine the systematic efforts implemented to address these factors, and provide a structured taxonomy that captures both educational and workplace interventions. Given the rising concerns of mental health challenges and work-related stress in STEM, it is crucial to identify gaps in current research and practice. To our knowledge no other existing literature reviews have collectively provided a taxonomy for technostress in these particular settings nor identified metrics related to technostress within software design. By reviewing these factors and interventions, this chapter seeks to highlight open challenges, inform educators and employers, and outline future research directions that can improve mental health outcomes and wellbeing for STEM learners and professionals.

A major contribution of this chapter is the development of a comprehensive taxonomy that classifies and connects the key constructs, relationships, and themes across the three themes. Constructed through a systematic approach, the taxonomy provides a structured map of the current knowledge landscape. It identifies conceptual overlaps, research trends, and gaps in understanding, highlighting areas for further investigation. The taxonomy is used to derive the second contribution of this chapter: a set of technostress-specific metrics for measuring the impact of software design decisions on user technostress levels (to be addressed as part of RQ1). These metrics form the core of a methodological framework proposed in Chapter 3 aiding in empirical investigation of software design decisions from a technostress lens. The taxonomy synthesises existing insights and identifies gaps in research, while the derived metrics operationalises these insights into a measurable form, providing a practical tool to inform the contributions presented in the upcoming chapters. Together, the taxonomy and the derived metrics represent the core contributions of this chapter.

2.1.1 Problem Statements and Research Questions

Chapter 1 identified technostress as a significant challenge within technology-intensive educational and workplace environments. However, despite a growing body of research across mental health, technostress, and AT, the literature remains fragmented, making it difficult to identify the factors, outcome measures, and intervention approaches that have been investigated across these domains. Furthermore, there is currently no unified structure for synthesising and categorising research that intersects these themes, limiting the ability to identify trends, relationships, and gaps that can inform future research and practice. This chapter addresses these issues through a structured review and taxonomy development process.

(RQ 1) What are the primary factors and outcome measures used in the existing literature to characterise technostress and its impact in STEM contexts?

(RQ 2) How can existing research that intersects mental health, technostress, and AT be systematically synthesised and categorised? Where are the key gaps in current research that a taxonomy can help identify?

2.1.2 Contribution and Significance

(C1) *Technostress-focused Taxonomy (addressing RQ2)*: This chapter introduces a systematically collated taxonomy that identifies conceptual overlaps, research trends, and gaps in understanding, highlighting areas for further investigation in the intersecting themes of technostress, mental health and AT in STEM contexts. The focus on the intersection between these 3 themes allows for a broad and deep understanding of technostress primary and secondary research.

(C2) *Technostress-specific Derived Metrics (addressing RQ1)*: The study introduces a set of novel metrics based on the literature and taxonomy, integrating technostress, mental health, and AT factors. While these outcomes are individually common in the literature, our contribution lies in (i) treating them as an integrated set of technostress-derived metrics in STEM contexts, and (ii) embedding them systematically in the intervention framework in Chapter 3.

The novelties of the contribution are as follows:

(N1) *Unexplored Intersection*: By examining the intersection of technostress, mental health, and AT in STEM, the study identifies underexplored areas (specifically the lack of technostress-specific metrics and systematic approaches that can be used to evaluate interventions supporting STEM learners/professionals) in both education and professional settings. Addressing these gaps lays the groundwork for frameworks and interventions that can be leveraged to better support learners and professionals in technology intensive environments.

2.2 Structure

The remainder of this chapter is organised as illustrated in Figure 2.1. It begins with an introduction to the proposed taxonomy, followed by an overview of the foundational themes and motivation underpinning the chapter. The inclusion and exclusion criteria used to select the literature are then presented to establish the scope of the taxonomy. Next, the findings from the literature search are discussed, highlighting key patterns, relationships, and research gaps across the identified themes within educational and workplace contexts. This is followed by a discussion that interprets the significance and implications of these findings in relation to existing research. The chapter concludes with a summary of the key findings and a discussion of the internal and external threats to validity, paving the way for the contribution presented in the following chapter.

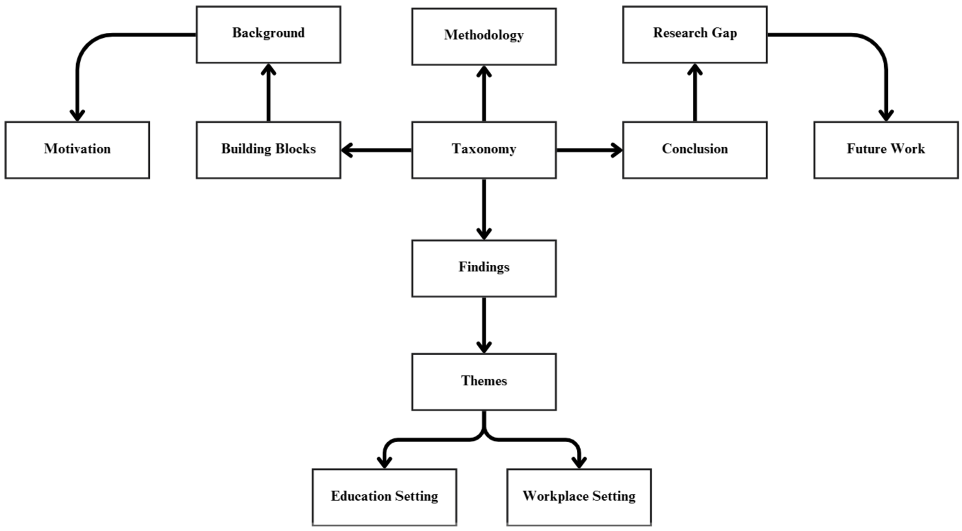


Figure 2.1: Chapter 2 Structure

2.3 Building Blocks

2.3.1 Background

The present chapter focuses on the three afore mentioned interrelated themes (Figure 2.2): mental health, technostress, and AT. Mental health forms the primary lens because it directly shapes how individuals perceive, respond to, and interact with their environment. Technostress is included due to its growing relevance in digitally intensive STEM environments, where Information and Communication Technology (ICT) has become both a necessity and a source of strain (aiding in addressing RQ1). This study broadens the scope of AT and incorporates it as a potential intervention pathway, offering tools to mitigate stress, support cognitive functioning, and promote emotional resilience. Together, these themes represent the intersection of human factors, technological environments, and support mechanisms, an intersection critical for understanding existing interventions related to technostress and mental health and thereafter the existing gaps in the research which is the core of RQ2.

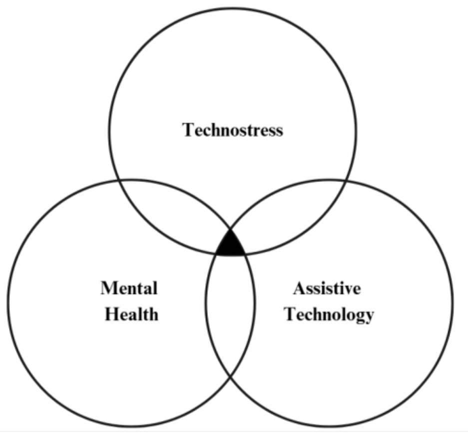


Figure 2.2: Taxonomy Building Blocks

2.3.1.1 Theme 1 – Mental Health

Historically mental health has mostly been understood in terms of mental illness. As [12] explain, society has often used the term mental health in everyday, academic, and scientific settings to mainly refer to negative states of mind. For over a century, psychological research has conflated mental health with mental illness, leading to the tacit agreement that mental health is solely defined by the presence or absence of suffering [13].

Consequently, researchers in psychology, health, and education have tried to define mental health in broader ways, moving the focus beyond just the absence of illness. From the shift of the focus on just the absence of illness, the World Health Organisation (WHO) has defined mental health as "*a state of well-being in which the individual realizes his or her own abilities, can cope with the normal stresses of life, can work productively and fruitfully, and is able to make a contribution to his or her community*" [14].

The two main traditions used to study wellbeing are the hedonic and the eudaimonic traditions [15]. The hedonic tradition is described as the "*presence or absence of negative affect*" and focuses on happiness, built around the idea of subjective wellbeing [16]. In this view, wellbeing is seen as subjective because it relies on people evaluating their own sense of wellness and interpreting their personal experiences [17]. By contrast, another perspective argues that wellbeing is more than simply being happy, as people who report happiness or life satisfaction may not always be psychologically well. This line of thought led to the eudaimonic tradition, which centres on living a meaningful and deeply fulfilling life [18]. It "*equates mental health with human potential that, when realised, results in positive functioning in life*". Wellbeing here is not treated as an outcome but as a continuous process of becoming one's true self, developing virtues, and living in a way that aligns with personal purpose. Some researchers also highlight overlap between the two traditions. For example, [16] suggest that adopting a eudaimonic way of life naturally brings hedonic pleasure, while [19] argue that not all experiences of pleasure stem from eudaimonic living.

The authors in [20] argued that although the WHO's definition of mental health marked a shift away from focusing solely on illness, it remains problematic. Linking mental health too closely with positive feelings and functioning, overlooks the reality that individuals with good mental health may still experience sadness, anger, or other negative emotions. Similarly, someone unable to work could be unfairly judged as lacking mental health. To address this, [20] proposed a more inclusive definition that avoids restrictive or culturally specific assumptions, framing mental health as a dynamic state of internal equilibrium. This perspective recognises the importance of emotional regulation, empathy, cognitive and social skills, and the capacity to cope with adversity. In a complementary view, [21] emphasised the need for cultural sensitivity in defining mental health, reinforcing the importance of an approach that reflects the diverse and multifaceted nature of human experience.

These broader perspectives highlight the importance of understanding mental health as a complex, dynamic state, rather than simply the absence of illness. In its 2022 report, the WHO identified mental health conditions as a "*growing crisis*", affecting approximately one in eight people worldwide equivalent to 12.5% of the global population. [22] conducted a longitudinal study involving 350,000 diverse undergraduates nationwide and found that various mental health conditions have increased drastically between 2013 and 2021. Despite increasing awareness and investments in prevention and intervention strategies, there has been no global decline in the prevalence of mental health problems since 1990 [23]. The implications are substantial, as mental health disorders consistently rank among the top ten leading causes of disease burden worldwide. These figures may underestimate the true scope of the issue, as stigma, fear of discrimination, and reluctance to disclose symptoms often discourage individuals from seeking professional help [24].

2.3.1.2 Theme 2 – Technostress

In the psychological literature, stress refers to a transaction between a person and his/her environment, in which an individual's appraisals of demands, events, or situations, as well as their coping resources, are considered central to whether an intense reaction will occur [25] [26].

Of particular relevance to this chapter are the IT-related elements which influence mental health [27]. The literature on stress related to software usage tends to focus on the phenomenon of technostress. Technostress was first coined in 1984 by American psychologist [8], who described it as "a modern adaption disorder brought on by the difficulty of coping with new computerised technologies in a healthy manner" [28], or as put by [29] "*stress experienced by end users in organisations as a result of their use of ICT*". The negative consequences of technostress include increased anxious and depressive symptomatology, decreased motivation, and intention to quit workplace, study and/or routine activities [30].

Technostress research is framed through the Person-Environment (P-E) fit theory [31]. The P-E theory focuses on the fit between an individual and their surrounding environment. It suggests that stress does not arise solely from the person or the environment but emerges when there is a misalignment between the two within a complex, multidimensional context [32].

Focussing on STEM educational and workplace contexts, technostress can have a negative impact sometimes leading individuals to disengage from STEM fields altogether [33] [34] [9]. Poorly managed technostress can result in greater resistance to adopting new technologies [35] thus contributing to individuals leaving STEM careers. This highlights a clear gap, the need to design these technologies in a way that reduces such resistance, forming the core motivation behind this research and the contributions presented throughout the subsequent chapters.

Beyond contributing to individuals leaving or avoiding STEM careers, technostress is increasingly being studied for its impact on overall health, with growing evidence suggesting it can harm holistic wellbeing [36] [37] [38]. Physical responses to technostress include elevated blood pressure, increased stress hormone levels, and a higher heart rate [35], whereas emotionally, technostress may lead to exhaustion, fatigue, anxiety, and anger, which can contribute to depressive thoughts [9]. Thus, mitigating technostress can support the retention of workers and students in STEM while also reducing associated health risks, a need that has become even more pressing as ever evolving computer technology forms the backbone of modern STEM with the COVID-19 pandemic further intensifying day-to-day reliance on it [39].

2.3.1.3 Theme 3 – Assistive Technology

AT encompasses hardware and software solutions designed to assist humans [40], primarily People with Disabilities (PwD) with little considering People not with Disability (PnwD) [41]. Not only are ATs one of the four pillars supporting global health, but they are also a critical element for the improvement of functioning by individuals with age- and health-related disabilities [42].

The WHO defines AT as an umbrella term for technologies that support an individual's senses, communication, mobility, and decision-making across different areas of life [43], which as a result enables to overcome challenges and carry out tasks more efficiently and independently [44]. The International Organization for Standardisation (ISO) sees AT as products that improve the functions of persons [45]. According to the Assistive Technology Industry Association (ATIA), AT refers to any tool, device, software, or product designed to improve, sustain, or enhance the functional abilities of individuals with disabilities [46]. The Institute of Electrical and Electronics Engineers (IEEE) defines AT as anything that helps a person improve performance, enhance function, or gain faster access to information [47]. In summary, AT can be described as any tool, system, or service that supports individuals (primarily PwD) in improving performance, enhancing function, or enabling quicker and easier access to information. AT can be classified into low-tech and high-tech solutions. Low-tech AT refers to simple, non-electronic tools such as magnifiers or pencil grips that require no programming, whereas high-tech AT encompasses more advanced, programmable devices like computers. Experts like [48] and [49] differentiate these two categories, highlighting the distinction between straightforward, easy-to-use tools and more sophisticated technological solutions. Low technology is often inexpensive and simple to develop or obtain, necessitating a streamlined operation method. High technology includes electronic components.

AT has received growing global attention in recent years for its role in promoting health, enhancing wellbeing, and improving quality of life. According to the WHO, an estimated 2.5 billion people, approximately one in three worldwide require at least one assistive product to support daily functioning and independent living [50]. However, only one in ten people have access to the AT they need. With more chronic health problems and an ageing population, the demand for AT is expected to rise to 3.5 billion by 2050 [43].

2.4 Methodology

2.4.1 Defining our Terms

The language of mental health can often be shifting, nebulous and confusing. It is not our intention to attempt to resolve this problem here or to offer absolute definitions. However, it is important that there is clear distinction about what is meant when using each of these terms (as described in Table 2.1). Alternative definitions which may be more appropriate, helpful or accurate on other occasions are accepted.

Table 2.1: Taxonomy Terms

Term	***Definition***
Mental Health	Refers to a full spectrum of experiences ranging from good mental health to mental illness. At one end of the spectrum is a state of mental healthiness, typically defined as a "state of well-being in which the individual realises their own abilities, can cope with the normal stresses of life, can work productively and fruitfully, and is able to make a contribution to their community" [51]. At the opposite end of the continuum is mental illness, which refers to a diagnosable psychological disorder characterised by dysregulation of mood, thought, and/or behaviour [52].
Wellbeing	Encompasses a wider framework, of which mental health is an integral part, but which also includes physical and social wellbeing, in which optimum wellbeing is defined by the ability of an individual to fully exercise their cognitive, emotional, physical and social powers, leading to flourishing.

In addition to mental health and wellbeing, this study also relies on terms related to the educational and professional settings in which the research is situated. For clarity, the following definitions are adopted (as described in Table 2.2).

Table 2.2: Taxonomy Terms Continued

Term	***Definition***
Student	A learner enrolled in an educational institution, actively acquiring knowledge and skills in various subjects through instruction and assessments.
Employee	An individual who works for an organisation or employer in exchange for compensation.
K-12	A learner aged 5 to 18 before entering higher education, encompassing both primary and secondary schooling.
Higher Education	A term used to describe the institutional providers of undergraduate level education.
Workplace	A physical or virtual location where employees or individuals carry out their professional activities, tasks, and responsibilities.

2.4.2 Databases and Search Terms

The databases referred to in this chapter are Scopus, Science Direct, IEEE Xplore, ACM Digital Library, SpringerLink, Google Scholar, Researchgate, NCBI. The search strings used were "mental health", "technostress", "assistive technology", "STEM", "student[s]", "undergraduate[s]", "higher education", "employee[s]", "employment", "workplace", "organisation[s]", along with Boolean operators such as "OR", "AND". "AND NOT", to further narrow down the results. Snowballing (which refers to using the reference list of a paper or the citations to the paper to identify additional papers) was also applied by searching the previously discovered works for pertinent references in order to find other potential works [53]. With journals, conferences and workshop publications being considered. Due to some overlap between the search engines and databases' databases, duplicate papers were suppressed from the results.

2.4.2.1 Context

The scoping review had no geographical constraints and included studies on the themes of interest. This approach ensured that the findings were not limited to a particular cultural or regional setting, enhancing the breadth and applicability of the review. However, only studies published in English were considered, which may have introduced bias towards studies conducted in Western contexts is acknowledge. This decision was made due to practical resource constraints, including the unavailability of translation services, which may have led to the exclusion of potentially relevant evidence published in other languages. A total of 159 papers were examined across 1994-2025.

2.4.2.2 Types of Sources

The scoping review included peer-reviewed journal articles, conference proceedings, and relevant grey literature (e.g., government or organisational reports), drawing on the same databases and search terms outlined in Section 2.4.1. Both empirical studies (quantitative, qualitative, and mixed methods) and theoretical or conceptual papers were eligible for inclusion, provided they addressed themes related to mental health, technostress, or AT.

2.4.3 Inclusion Criteria

To ensure relevance and focus, this review adopted a set of inclusion criteria grounded in the study's research questions.

2.4.3.1 Educational Setting

+ Studies addressing mental health (theme 1) of students are included.
+ Studies examining technostress (theme 2) among students are included.
+ Studies on assistive technology (theme 3) use by students, including PwD and PnwD, are included.
+ Studies on themes of interest irrespective of their age group (K-12 AND/OR HE)
+ Studies on themes of interest irrespective of their gender.
+ Studies on themes of interest within STEM subjects.

2.4.3.2 Workplace Setting

+ Studies on themes of interest irrespective of their age group.
+ Studies on themes of interest irrespective of their gender.
+ Studies that mention themes of interest related to technology usage regardless of the industrial domain in which the technology is being used.

2.4.4 Exclusion Criteria

A set of exclusion criteria was established to filter out studies that fall outside the scope of this review. These criteria were applied to ensure that irrelevant, duplicate, or insufficiently detailed publications were not considered.

2.4.4.1 Educational Setting

- Studies focusing on themes of interest of HE professionals rather than students were excluded, as students are the primary targets of this research in the educational settings.

2.4.4.2 Workplace Setting

- Studies outside the scope of workplace/organisational environment (e.g. domestic use of technology, freelance work, etc).

2.4.4.3 Health Outcomes

- Studies focusing exclusively on physical health outcomes without examining cognitive, emotional, behavioural, or psychosocial aspects of technostress were excluded. While technostress can manifest through physical symptoms, the primary focus of this review is on understanding the cognitive, emotional, and psychosocial mechanisms that contribute to technostress and can be addressed through technology design and intervention strategies.

2.5 Findings

The purpose of this section is to review and synthesise existing research on mental health, technostress, and AT across both educational and workplace settings. In doing so, it not only maps the scope of current knowledge but also highlights key findings and presents the proposed taxonomy.

2.5.1 Mental Health

2.5.1.1 Educational Setting

There has been a sixfold increase in student mental ill health since 2010, resulting in a steady growth of research on mental health and well-being in university students over the last decade [54], [55]. Young adults today are more likely than earlier generations to encounter mental illness, findings by [56] reveal an increase in the prevalence of common mental disorders (CMDs) for those aged 16 to 24, which has a knock-on effects on the quality of life in university students [57]. Given that the onset of mental health problems typically occurs before age 24, university students particularly those in their second and third years show higher levels of mental health difficulties compared to first-year students [58]. Given the male dominance within the STEM field, male STEM students are at more risk of these mental health issues [59], [60], [61].

Performance: Academic failure in higher education is a complex issue that requires consideration of multiple factors. In recent years, growing attention has been directed toward understanding students' academic performance [62]. Academic performance is a significant source of stress for many university students [63], [64]. The StudentLife sensing app by [65] evaluates the effects of workload on stress, sleep, activity, mood, sociability, mental health, and academic performance on a daily and weekly basis. The study statistics indicate that students begin the term with high levels of positive mood and discourse, low levels of stress, and normal patterns of daily activity and sleep. Stress significantly climbs as the semester goes on and the workload grows, while good mood, sleep, conversation, and activity levels decline. This data proved invaluable in situations where professors intervened to support students who had missed several weeks of lectures and failed to submit multiple assignments. As a result of this timely intervention, those students were able to complete their coursework over the summer and avoid failing grades.

Stress/Anxiety: Mental health, especially anxiety, has a significant impact on academic performance. While stress typically arises in response to external pressures, such as exams, anxiety is an internal state that can persist even in the absence of immediate challenges [66]. High anxiety correlates with lower academic performance [67] though moderate anxiety levels may be optimal [68]. This is consistent with the Yerkes–Dodson Law, which suggests that moderate levels of arousal improve performance, whereas very low or very high levels hinder it [69]. Depression is another prevalent mental health condition linked to poorer academic performance [70], often marked by ongoing sadness and a diminished interest [71].

Success: The authors [72] investigated the relationship between academic success and mental health issues among university students. Findings show that the rise in mental health issues lowers university students' academic performance, which has a knock-on effect and leads to negatively affecting both professional and personal lives of students. This was further supported by [73], their findings also found associations between academic performance and aspects of mental health.

Sleep: Sleep also plays a crucial role in academic performance, with students who experience poor quality, insufficient, or irregular sleep often showing lower academic achievement [74]. Despite recommendations for seven to nine hours of sleep [75], nearly one-third of students do not meet this guideline [76]. As a present study by [77] found there are gender differences, where for male students, interventions should emphasise sleep awareness, whereas for female students, priority should be placed on mental health awareness.

Degree: While some university students are able to handle increased stress or obstacles while maintaining their academic success, others are unable to manage the stress, which negatively impacts their academic performance as highlighted by [78] thus causing them to drop out of school without earning a degree [79]. Undergraduate students experiencing poor mental health in their first semester were found to be at a significantly higher risk of reduced academic performance throughout their studies [80]. Similarly as [81] found, there is a significant relationship between students' mental health status and their academic performance, showing that students with severe mental health issues are more likely to repeat courses compared to those with mild or normal issues. As [82] further highlights, improved well-being was consistently associated with better academic performance.

Financial: The increasing participation of young people in higher education over recent decades has drawn greater attention to student mental health and well-being, while also highlighting concerns about the impact of financial stress. For instance, in England, rising tuition fees and the replacement of grants with loans have resulted in the average undergraduate leaving university with debts of approximately £50,000 [83]. While financial aid is widely recognised as a means of supporting students in need, there remains limited understanding and research on how to efficiently identify those individuals. Traditional approaches that rely on advisers' subjective assessments are often ineffective and may not accurately reflect students' real circumstances. [84] addressed this gap by developing a framework called Dis-HARD, which determines students' eligibility for financial aid by analysing their complex on-campus behaviours through digital footprints—the traces left as students interact within cyber-physical environments. This framework predicts the optimal portfolio of stipends that should be allocated to each student.

Wellbeing: Findings by [85] also suggest a significant link between students' financial circumstances and their mental health, as well as their perceptions of those circumstances given the current higher education context of rising tuition fees, growing debt, and increased use of university mental health services, students' well-being was found to be closely tied to financial factors. Those experiencing poorer well-being were more likely to face difficulties due to personal debt, low household income, reliance on bursaries, or lack of parental financial support. Findings by [86] indicate that financial stress functions both as an independent stressor and as a compounding factor, intensifying existing vulnerabilities while contributing to emotional distress and a diminished quality of life.

Stress: Work by [87] found no connection between student stress levels and debt (measured in expenditure terms). They stated that student loan debt might now be viewed as a normal aspect of university life. However, other research by [88] highlights that rising tuition costs, heavier debt loads, and the challenges of managing personal finances for the first time contribute to financial stress, which has been linked to poorer mental health, increased anxiety and depression as [89] found, reduced academic performance and even dropouts, concerns all relevant not only to educators and administrators but also to employers. [90] reported that debt-to-income ratio and attitudes towards debt were not significant predictors of stress. Instead, financial well-being (which reflects an individual's overall sense of financial security, ability to meet expenses, and confidence in managing money) was found to be more strongly associated with stress outcomes.

Performance: [91] found that students experiencing poorer mental health were more likely to report that financial worries negatively influenced their academic performance. Moreover, those who acknowledged that such concerns affected their studies generally achieved lower performance outcomes compared to students who did not report this impact. As [92] found, working more than 14 hours per week had a negative impact on students' academic performance.

Degree: According to [93] students with greater financial concerns also reported higher levels of stress, anxiety, sleep difficulties, and feelings of being judged by others. These pressures intensified over time, with mental health indicators worsening in the third year compared to the first and second as found out, reflecting the cumulative burden of debt and heightened concerns about repayment after graduation. [94] conducted analyses demonstrating that the level of financial burden affected mental health, partly through increased consideration of abandoning university and the necessity of working longer hours.

Curriculum: In a qualitative study by [95], students reported that curriculum processes, including teaching, assessment, and academic support contribute significantly to wellbeing. As [96] outlined, students identified the curriculum as a key factor influencing their ability to engage with their studies. It not only shaped how they coped with challenges but also affected their motivation, often leading to negative emotions. Practices such as scaffolded challenge and appropriate assessment methods were identified as supporting wellbeing. As [97] study highlights, efforts to redesign a curriculum were appreciated and added value when the students' voices were heard, further suggesting space for more inclusive curriculum design with one dimension of inclusivity being mental health awareness specifically through designing more mental health aware technologies to enhance curricula. A study by [98] showed that reducing the weekly in-class curriculum from five days to four days significantly improved mental and physical health, suggesting that workload and pacing in the curriculum affect wellbeing. The study by [99] positions the curriculum as central to improving student mental health – examining how curriculum content can be leveraged to promote mental health awareness through three key approaches: fostering interdisciplinary connections, adopting a curriculum-infusion model, and drawing upon specific disciplinary knowledge bases.

Assessment: Riva et al. [100] analysed how student-centred learning and assessment can serve as enablers of student wellbeing, fostering a sense of value, engagement, and recognition. However, their study also emphasises that poorly planned or executed student-centred approaches may negatively affect wellbeing. Achieving openness, activeness, and engagement requires striking a careful balance: avoiding imposed participation while ensuring clear, flexible support and feedback. According to [101], there has been limited investigation into how evaluation strategies in UK higher education affect student wellbeing. Assessment is often perceived as the "first point of failure," with students emphasising its impact on their academic experience and mental health. Their findings suggest that providing students with greater autonomy over assessment design can enhance their sense of individuality and ownership. While some students experience empowerment and control, others, particularly less confident learners, may find such freedom stressful. Similarly, balancing personalised, detailed feedback with the practical demands of high student enrolment and intensive grading presents challenges for staff, creating additional strain on resources. As [102] highlights poorly designed assessment systems can frustrate students' psychological needs, leading to reduced motivation and poorer psychological outcomes. Conversely, when assessments support these needs, wellbeing is enhanced. Building on this, studies [103], [104] found incorporating self-assessment as part of assessment tasks promotes student more independence and a stronger sense of control, which in turn can enhance engagement and reduce stress. The Computing: Ipsative Assessment (C:IA) tool developed by [105] helps to address the problems that could be creating potential student unhappiness with assessment and feedback in UK higher education. The web-based system focuses on goal-setting and progress monitoring for students. The system received a largely positive response, with particular praise for the personalised features. When students are inspired to participate in learning activities, they learn more effectively. As [106] explored the use of an e-assessment tool which provides students with a unique learning environment while also assisting university academics in managing the growing number of students with limited resources. The e-assessment tool provided students with several submission opportunities and response to their questions. Results showed an increase in pupils' interest in learning, as they were motivated to fix their mistakes and gain rewards. The academic performance of the students was enhanced by providing them with the opportunity to correct their mistakes, focusing on the aspect of personalised learning environments (PLEs), through providing students with constructive criticism. Consequently, this led to the development of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) by [107] which utilises a rule-based system that considers the features of both the assessment question and the student answer. The findings showed that students with little prior knowledge dramatically improved their performance after receiving tailored feedback.

Deadline: A study by [108] highlights that students report deadline bunching (many deadlines happening at once) causes them to feel overwhelmed. Also, when students don't have information early enough to begin work, deadlines get closer together, increasing stress and reducing wellbeing. What tends to happen as a result is students prioritise academic tasks over wellness. Many feel that overlapping assignments, tight schedules, and lack of support during "deadline periods" negatively affect their sleep, motivation, and mental health [109]. High-pressure academic periods (e.g. exam weeks), also suggests deadlines and assessment periods contribute to increased stress [110]. A study by [111] similarly states students showed elevated stress around exams; among underrepresented minority students, greater stress is associated with better performance, but still reflects the pressure associated with assessments and deadlines.

Tools: Studies have explored students' perceptions of their learning environment [112] and its relationship with their mental wellbeing [113]. Such tools often referred to as learning management system (LMS) [114] or virtual learning environment (VLE) and are a web-based environment commonly used to support learning. Its web-based design enables students to access VLEs anytime and from anywhere [115]. Research on comparisons of various LMSs is rare but some comparisons between Moodle, Blackboard, Canvas, and others are available in the literature [116]. As [117] highlights such tools like Moodle are mainly used within University STEM disciplines and can effectively improve student performance, satisfaction, and engagement. Moodle is increasingly being used as a platform for adaptive and collaborative learning and used to improve online assessments.

2.5.1.2 Workplace Setting

According to WHO [118] 15% of working-age adults are thought to be suffering from a mental illness. It is estimated that on an annual basis, up to £56 billion is lost annually by UK companies due to poor mental health as a result of increased absenteeism, burnout and turnover, lower productivity and performance [119]. Workplace stress has been defined by the WHO as a "pattern of physiological, cognitive and behavioural reactions to some extremely taxing aspects of work content, work organisation and work environment" [120].

Creators: In the workplace, several factors are recognised as key determinants of employees' mental health [121]. These include high job demands, limited job control, low social support, imbalances between effort and reward, lack of procedural and relational justice within the organisation, organisational change, job insecurity, temporary employment, irregular working hours, bullying, and role-related stress [122]. A study by [123] which looked into the impact of mental health on wellness in adult workers revealed that depression, stress, and anxiety were found to reduce wellness, while self-efficacy, overall well-being, and perceived health status contributed to improving it. Such factors can have knock-on effects as [124] found, leading to early retirement, disability, and absence from work are all disproportionately impacted by mental health illness. When people experience mental health issues, they are more likely to experience longer periods of unemployment, become disabled, and retire earlier. [125] employed ML techniques to predict employee burnout rates using a curated set of contributing factors, addressing the issue that many employees remain unaware they are experiencing burnout, leaving symptoms untreated and potentially leading to long-term problems. [126] presented a chatbot which assess employees' mental health in a cost-effective and engaging way and provide useful information for both the employee and the employer. Based on the results the system proved to be very engaging and successful in collecting anonymous mental health data from employees, as they reported symptoms of anxiety, stress, depression and burnout.

Interventions: As [127] discusses, disclosure allowed workers to build positive workplace identities, presenting themselves as authentic, responsible, and trustworthy. It also alleviated the stress associated with concealing their identity and allowed them to position themselves within the broader context of mental health in the workplace. However, for intervention to succeed as [128] points out, a lack of engagement, whether from the employees as a whole or from specific employee groups, was often identified as a key reason why interventions failed or did not achieve the expected outcomes. This theme is widely recognised as a critical factor in the success of workplace health promotion initiatives. Findings from [129] suggest that certain intervention characteristics have the strongest impact. These include combining multiple therapeutic approaches such as, integrating technology to reduce dropout rates, and delivering interventions in group formats to engage a larger number of employees. As [130] highlights, instead of responding only after cases of mental ill health arise, employers can be more effective by taking a proactive approach, valuing employees as key assets, assessing psychosocial risks, and actively promoting health and wellbeing. [131] identified the eight categories of best practices for supporting mental health in the workplace. [132] presented a framework for an integrated approach to workplace mental health interventions. This brings together "*harm prevention*" addressing workplace organisation primary prevention initiatives, "*positive mental health promotion*" addressing individuals' resilience in mitigating effects, and illness management through diagnosis, treatment, and reintegration. Knowing if employees require intervention to be put into place is key, hence why [133] explored the development of classification methods for assessing whether or not an employee needs mental health care. [134] employed Machine Learning (ML) algorithms to identify factors contributing to mental stress among employees, finding that the Random Forest algorithm achieved the highest precision and accuracy at 75.13% in detecting stress levels. [135] similarly applied machine learning techniques to predict mental health disorders using various attributes, concluding that a personal history of mental health disorders was the most influential factor in prediction accuracy, followed by a family history of such disorders. A study by [136] used the Perceived Stress Scale (PSS) technique for stress prediction, along with an range of ML algorithms to predict whether or not a person is stressed. With an accuracy of 99%, precision of 0.98, and sensitivity of 1, they were able to identify that amongst data captured from 251 employees only 9.6% were stress free.

Outcomes: Symptoms of mental illness especially when hidden, have the capacity to impact negatively on employee work performance and/or attendance [137]. A review by [138] states that poor mental health is associated with negative work outcomes and highlighted that it leads to poorer mental health, which consisted of absenteeism, presenteeism., affecting company turnover, causing longer term disability, affecting interpersonal conflict and sustaining physical illness/sickness. An analysis of the happy-productive worker by [139] reveals a definite correlation between emotional wellbeing and work. When compared to unhappy employees, cooperative employees are more helpful to their co-workers, more punctual and time efficient, show up for more days of work, and stay with the company longer [140]. Positive feelings at work have been linked to creativity, innovation, and innovative thinking [141]. As [142] evaluates, tackling mental health in the workplace can enhance productivity of the business, improve employee health, improve employee performance and improve community well-being.

2.5.2 Technostress

2.5.2.1 Educational Setting

In today's era of rapid technological advancement, the integration of technology into education has become widespread, offering the potential for richer learning experiences and improved academic outcomes [143]. The constant exposure to digital stimuli can cause anxiety, distraction, and cognitive overload, which in turn hinders students' ability to focus effectively on their academic work [144]. Additionally, many students find it challenging to stay abreast of the latest technological developments and adapt to new digital tools and platforms in academic settings. The ongoing demand to learn and master these technologies can lead to feelings of frustration, inadequacy, and anxiety, especially among those with limited technical knowledge. Moreover, the rapid pace of technological innovation, coupled with frequent software and hardware updates, further exacerbates technostress among students [145].

Experience: According to [146], students with lower ICT expertise, as well as postgraduates and older learners, reported higher levels of technostress, which negatively influenced their academic productivity. Among the five dimensions of technostress, techno-uncertainty was found to be the least distressing. In contrast, techno-invasion emerged as the most significant source of stress for students. Techno-complexity was also identified, particularly among younger students who perceived technology as difficult to navigate and manage. similarly [147] also found that students with a higher level of experience, higher confidence in computer usage and lower anxiety are better equipped to deal with technostress. Interestingly, [148] study found that older students experienced less techno-complexity, while greater use of information and communication technologies (ICT) was linked to higher techno-uncertainty. Gender also played a role, with male students reporting stronger perceptions of techno-invasion.

Creators: A conceptual framework by [149] outlined that techno-overload, techno-invasion, techno-complexity, techno-insecurity and techno-uncertainty as all being linked and negatively affecting academic performance, as a result of digital literacy, accessibility and lack of ICT skills. A study by [150] supports this view, identifying digital literacy as a major factor in reducing technostress. Their findings indicate that technostress negatively affects academic performance and productivity, with key contributing factors being techno-complexity and techno-invasion. Poor academic outcomes often arise when ICT tools designed to support learning are overly complex or difficult to use. Conversely, students with higher levels of digital literacy experience significantly lower levels of technology-related stress. The study concludes that improving digital literacy can substantially reduce technostress, particularly that caused by technology's complexity and insecurity. A study by [151] highlights techno-uncertainty and techno-invasion were identified as the most significant technostress creators, driven by frequent software and hardware updates and the additional time required for students to adapt to new ICT systems. The study also emphasises that system accuracy and output quality are key to end-user satisfaction. Stress, anxiety, and depression can result from coping with the complexity of technology and/or the unpredictability that comes with ongoing innovations, upgrades, and changes in ICT. Offering students accessible ICT services, training, workshops, as well as detailed online ICT instructions and resources can enhance techno-ease and problem-focused solutions [152]. Being able to effectively utilise digital tools the impact of technostress on academic performance and productivity can be reduced as found [153]. A study conducted by [154] unveiled a significant negative relationship between technostress and psychological health. As the dark side of ICT usage, technostress negatively impacts students' academic productivity [155] and even leads to learning burnout in Technology Enhanced Learning (TEL) [156]. As such, [157] found that the multidimensional P–E misfit of technostress did not directly influence students perceived performance but had an indirect effect through burnout in TEL. This aligns with the understanding that stress, including technostress, typically operates through intermediary processes before impacting cognition. Technostress on multiple dimensions of P-E misfit often firstly negatively impacts individuals' well-being, leaving students frustrated, exhausted, and overwhelmed (i.e., experiencing burnout), which in turn diminishes cognitive functioning and lowers academic performance [158]. Similarly, [159] further backs this as their findings indicate that technostress creators, including techno-complexity, techno-insecurity, and techno-uncertainty, were significantly associated with students' burnout in TEL.

Outcomes: According to [160] technostress plays a key role in shaping students' motivation within higher education. Such impacts not only affect students inside of class but outside of class too as explored by [161] ,who found that technostress weakens the link between facilitating conditions and Digital Informal Learning (DIL). In other words, even when students have good support and resources, the positive effect on their informal learning is reduced if their levels of technostress are high. [162] examined the relationship between student satisfaction and performance among university students and the aspects of technostress (techno-overload, techno-complexity, techno-insecurity, and techno-uncertainty) in relation to those factors. Their findings revealed that in comparison to techno-uncertainty, techno-complexity shows a more notable impact on student satisfaction and performance expectations. However, the findings expressly show that there is no substantial relationship between students' happiness/performance and techno overload and technological insecurity. The results of a study by [163] showed that student satisfaction facilitated the relationship between techno-complexity, techno-insecurity, and performance expectancy. Both techno-complexity and techno-insecurity negatively influenced student satisfaction, while student satisfaction positively impacted performance expectancy. These findings suggest that lowering levels of techno-complexity and techno-insecurity can enhance student satisfaction, thereby strengthening students' expectations of achieving better academic performance. As [164] found, technostress positively influenced exhaustion, while exhaustion negatively influenced academic performance. Focusing on the consequences of performance, [165] study shows, higher levels of technostress are related to absenteeism, missed deadlines and not achieving objectives. The knock-on effects of technostress can be significant in terms of also affecting sleep quality as [166] found, students with higher levels of technostress reported poorer sleep quality.

Interventions: As highlighted by [167], factors that shape academic success play a crucial role in students' overall growth, development, and sense of fulfilment throughout their educational journey. Elements such as academic motivation and psychological well-being are particularly influential in enhancing learning outcomes. As a study by [168] in K-12 highlights, applying time management and boundary-setting strategies, participants were able to regulate their digital interactions more effectively, which reduced feelings of overwhelm and boosted productivity. Overall, the intervention not only alleviated technostress but also promoted greater academic focus, healthier sleep patterns, and improved emotional well-being.

2.5.2.2 Workplace Setting

A major source of stress within organisational settings is technology, as employees are obliged to utilise several different ICT applications in order to complete their work tasks. The adoption of ICT in the workplace has increased recently thanks to the coronavirus disease 2019 (COVID-19) pandemic, which required remote and, consequently, digital working arrangements. A review by [169] on existing research reveals that, while some findings on resources and coping methods are available, no preventive approaches have been rigorously assessed. According to the transaction theory of stress [170], technostress creators are expected to increase strain for the employees, and technostress inhibitors are expected to decrease strain.

Experience: Younger employees, like millennials, who are accustomed to using various media on a regular basis may be less susceptible to technostress because they are digital natives, but older workers may have lesser levels of digital literacy [171]. A study by [172] examined millennials motivations for using social media and their level of happiness at work. Findings indicated that millennials utilised social media more frequently for work-related goals and reported higher levels of psychological discomfort, burnout, and technostress, despite the fact that they might be better technologically equipped. Conversely, the findings from [173] study suggest that older workers react more positively to implementation of IT initiatives than their younger counterparts, contradicting conventional beliefs that older adults resist IT innovation, however contrary to this, older workers may be more affected by overload, as they often face greater difficulties in mastering new technologies and have established workflows that are disrupted by technological changes.

Creators: A longitudinal study by [174] on government employees found that technostress creators harm employee wellbeing, causing greater physical discomfort and other health issues. The results of [175] study further show that technostress creators are bad for employees' work-life balance and have become important causes of stress, strain, worry, and pressure in organisations. The results of [176] longitudinal study showed that work exhaustion plays a mediating role between technostress and employee work well-being, meaning that higher levels of technostress can lead to exhaustion, which in turn negatively impacts overall well-being at work. An analysis by [177] found that that different forms of technostress arise under distinct conditions: techno-complexity is common when employees face high work demands, low personal control, and limited commitment to technology; techno-insecurity emerges when employees hold negative attitudes toward technology, seeing it as neither useful nor motivating, and instead as a threat to job stability; and techno-invasion occurs when individuals are required to use technology extensively in their work but lack both motivation and a sense of control, reflecting broader struggles in how they manage technology in their daily lives.

Outcomes: The findings of [178] study reveal that technostress mediates the link between workplace design and physical discomfort, underlining its impact on employee wellbeing. It highlights the importance of ergonomic improvements and supportive organisational practices to reduce technostress and discomfort, ultimately improving employee productivity and satisfaction. It has been supported that when employees are provided with adequate technological resources and support, their productivity increases as technostress is mitigated [179]. Organisational support and work–life balance policies help employees manage technostress factors like overload, invasion, and complexity, leading to better wellbeing and workplace flourishing, even in remote working contexts [180]. [181] results show that job satisfaction influences employees' work engagement when using ICT, and that both technostress creators and technostress inhibitors are important factors that affect engagement levels, both directly and indirectly, through job satisfaction.

Interventions: [182] findings indicate that both mindfulness and IT mindfulness help protect individuals from the negative effects of workplace stress caused by ICT use. More mindful individuals are better able to adapt and cope with daily technostress that arise daily due to the extended use of organisational ICTs, and higher levels of mindfulness can reduce its harmful emotional impact, while at the same time enhance user satisfaction and improve task performance. Similarly [183] also discusses, when faced with a stressful scenario, mindful people are more likely to use adaptive coping mechanisms, also the possibility that users may feel techno-stressors is decreased by mindfulness. It is also important to note, as [184] , points out, instead of requiring all employees to use technology, organisations could offer alternative workflows that allow staff to choose non-technological methods where possible or ensure that digital tools are designed with more options for personal control and customisation. In addition, organisations could introduce clear policies supporting the right to disconnect. Such measures would help reduce feelings of techno-invasion by giving employees more control over their boundaries and allowing them to manage work in ways that match their comfort levels. [185] proposed the task-technology fit theory aiming to support organisations in pinpointing factors affecting technostress in employees and interventions to mitigate them. While sharing the motivation with this study, the proposed mitigations do not focus on the design of the technologies used and it is more catered to workplace settings where software users have different experience levels from student developers. The study by [186] proposed a semi-automatic process designed to help employees cope with technostress and improve job satisfaction. The process begins with employees completing a survey-based questionnaire, which initiates the technostress detection phase that identifies one or more technostress creators affecting the individual. The subsequent coping phase integrates human involvement through mental health specialists responsible for monitoring and diagnosis, thereby combining automated detection with professional intervention to create a semi-automatic support mechanism. Another study also by [187] proposed an approach in which employees complete a technostress-based questionnaire. Once submitted, the data are processed through a Knowledge Management Server (KMS), which recommends appropriate coping strategies. Strategies that prove effective lead to increased employee satisfaction, while those that are ineffective are re-evaluated by repeating the process, creating an iterative system for managing technostress. [188] proposed a conceptual framework examining the impact of coping strategies on computer-related technostress among employees. Their study identifies several validated strategies for managing computer-related technostress. By effectively applying these coping mechanisms, employees are expected to use computer applications more productively, which in turn can enhance both individual and organisational performance.

2.5.3 Assistive Technology

2.5.3.1 Educational Setting

Over the past two decades, there has been a significant global rise in the diagnosis of mental disorders, alongside a rapid increase in the prevalence of various mental health problems [189]. Depending on the type of psychological disorder, students may face challenges with socialisation, communication, and adapting to environmental changes, which can hinder their ability to concentrate effectively [190] and also have an knock on effects on student outcomes. AT encompassing a range of specialised tools, enable students to access education more effectively and participate actively in the learning process, thereby enhancing learning outcomes and strengthening the overall educational system [191]. A systematic review by [192] found that AI and HCI are more commonly used as assistive technologies compared to other tools, and research supporting students with learning difficulties is more extensive in primary education than in higher education.

Inclusion: A systematic review by [193] revealed that AT increases the accessibility and inclusion of students with disabilities. The results of [194] review emphasise that advanced AT significantly promote inclusion by enhancing accessibility and active participation for students with mobility impairments, thereby strengthening their academic engagement and social integration. [195] also supports this stating the importance of AT for SWD and how it's a human right just as access to medical or other health services, and education. SWD need utmost support for them to acquire and access AT to enhance their participation in learning and contribution to societal development without unnecessary inhibitions. As AT promotes social interaction, curriculum access and the ability to express understanding, there is the potential for heightened inclusion in the classroom [196].

Engagement: According to [197] it was observed that the use of AT services helped SwD to not only perform better academically, but also promoting engagement and active participation in classes [198]. Another systematic review by [199] found that access to appropriate assistive technology enhances students' educational engagement, well-being, academic self-efficacy, competence, adaptability, and self-esteem, with those whose needs were fully met scoring significantly higher in areas such as stress management, time pressure, and class communication compared to those with unmet needs. The systematic review by [200] highlights that AT can foster academic engagement, social participation, and psychological growth for students with disabilities, offering educational, psychological, and social benefits. However, effective use of AT may be limited by factors such as insufficient training, device limitations, lack of external support, and the difficulty of managing multiple information sources, which can hinder full engagement in higher education. A review by [201] further highlights the importance of AT, showing that student engagement can be enhanced by ensuring equitable access to rapidly evolving tools. This requires addressing affordability, assessing the usability and suitability of existing technologies, and improving accessibility by challenging the social stigma surrounding their use. Research by [202] utilised Augmented Reality (AR) to support the education of students with learning disabilities (LD). Their proposed prototype, Interactive Textbook, was designed to function like a standard textbook but without any special markers or identifiers. When a student focuses on a section of text, the system overlays 3D visuals, sounds, and videos that visually and audibly enhance comprehension, thereby creating a more interactive and accessible learning experience. In fast-paced or advanced classes, students with visual impairments often struggle to keep up with lectures. To address this, [203] developed a portable Tablet-PC-based Note-Taker that allows students to note-take easily between classes without requiring lecturers to alter their teaching materials. This solution promotes active engagement in notetaking during lectures, offering a more interactive and independent learning experience compared to relying solely on audio or video recordings.

Performance: A study by [204] found that AT use is associated with higher academic achievement for the students using it but also highlighted challenges that hinder the successful implementation of AT. [205] measured performance and satisfaction before and after AT intervention and saw improvements in academic task performance across groups. A study by [206] described AT as the saviour of mathematics in HE helping enhance mathematics learning by enabling students to study at their own pace and in their preferred style, improving performance while also reducing negative attitudes toward the subject. The findings of [207] study reveal that

AT based training has a significant positive impact on the academic performance and employability of SWD. Their study recommends integration of AT based training into the mainstream education system to provide equal learning and employment opportunities. [208] proposed KidKanit, a calculation aid tool offering a novel approach to mathematics education for students with dyscalculia through AT. Their study showed improvements in students' performance from pre-test to post-test, increasing their chances of success and providing equal learning opportunities compared with their peers. Similarly, Word Prediction software, developed by [209], supports students with learning difficulties by predicting the words they intend to type, thereby enhancing their writing skills.

2.5.3.2 Workplace Setting

One of the largest obstacles for people with disabilities seeking employment continues to be employer attitudes. Nearly three-quarters (73%) of managers would be concerned that people with disabilities would struggle to perform the job, making them less likely to hire them. Employers can be concerned about racking up extra costs due to either missed productivity or having to pay for adaptations [210].

Inclusion: ATs are crucial in enabling the inclusion of differently abled people in the workforce because they provide them equitable access to the workplace and a level playing field [211]. Only a limited number of businesses have proactively hired people with disabilities, yet those that have report gaining more motivated, productive, loyal, committed, and satisfied employees. With access to ICT and AT in the workplace, these employees can perform on par with their peers, leading to higher retention rates [212]. AT enables people with disabilities to access the workplace and participate fully, fostering a sense of belonging and making them an integral part of the organisation [213]. However as [214] highlights, for it to be beneficial employees and effective for the employer, companies must have an in-depth knowledge not only regarding disability, AT, labour laws, reasonable accommodation, but also related to humanities like pedagogy and psychology. It should not be assumed that adults with acquired disabilities are inherently aware of available alternative options to traditional work, and as [215] findings indicated, participants in their study did not have specific knowledge of available work alternatives which could accommodate their health and accessibility needs. However, when presented with accessible work alternatives, the majority of participants increased their interest in learning more about those options.

Barriers: Even though assistive technologies are available to help with different accessibility needs, many people choose not to adopt them. This is often because the technology does not work well enough or is too expensive [216]. A study conducted by [217] found that there were low employment rates for people with disabilities, low levels of technology availability and utilisation of assistive technologies, and employment barriers for people with disabilities. Barriers included: (a) expenses associated with use of assistive technologies, (b) low skill level and job applicability, (c) stigma, with employers who are uncomfortable around workers with disabilities are likely to be hesitant to hire or work with them. Similarly [218] found that discomfort and pain, limited knowledge of the technology's features, and the complexity of the technology. The amount of time required for training, limited work time available for mastery, cost of training and limitations of the training provided, all as contributing factors to the usage of AT. [219] highlights a large number of workplace accessibility difficulties, including many issues in disability-friendly workplaces, and that overcoming these barriers through the adoption of AT can help break down communication barriers and make it easier for employees with varying abilities to communicate better, which can improve collaboration and increase productivity. [220] highlights the need for stronger cooperation between occupational health services, IT management, and technical experts to ensure disabled workers can access the assistive technology they need. It also stresses that acquisition processes should be clearer, more consistent, and widely understood. Another study [221] highlighted cost barriers and how limited access to ATs reduces the possibility of self-fulfilment of people with disabilities regarding productive work and impedes their well-being, thus leading to a lack of income, in turn, blocks access to more modern and high-quality ATs. It was highlighted that employees frequently avoided employing AT that attracted undue attention to their disability [222]. [223] study outlined various barriers, most noticeable being fear of disclosure as a key barrier, driven by concerns about discrimination and negative cultural attitudes. Misconceptions and mindsets among employers and co-workers were also highlighted as major obstacles, with participants stressing the need to challenge stereotypes and recognise the value and skills people with disabilities contribute beyond token placements. Successful deployment of AT requires thoroughly assessing workplace-specific infrastructure, employer expectations, and usability factors [224]. [225] study found that several factors influence the use of technology accommodations, such as employer and coworker attitudes, employer knowledge, cost concerns, workplace policies, employee confidence, education, and self-advocacy skills. These findings are consistent with those of a systematic review by [226] on workplace accommodations for employees with physical disabilities. [227] study explored the perspectives of employed users, highlighting both the barriers they faced in using AT at work and the strategies that supported them. It further emphasised the importance of empowering AT users to manage their ongoing needs by providing knowledge of available resources and ensuring ready access to them, while also calling for workplaces to be proactive and establish clear processes to address these needs.

Outcomes: As [228] found, while accommodations of AT cannot address every employment need of individuals with physical disabilities, they are often essential for enhancing performance and achieving successful outcomes that meet both employee and employer expectations. [229] presented an evaluation framework and found that employees were generally successful in using AT devices and reported high productivity, but they also faced barriers such as limited technical knowledge and training, device reliability and compatibility issues, and accessibility constraints. A SLR by [230] outlined that the usage of AT showed enhanced accuracy and task completion rates, increased independence, and improved generalisation abilities among employees with cognitive disorders. Similarly, [231] also found that the use of AT can lessen an individual's dependence on a personal assistant while at work. A study by [232] further backs the usage of AT stating, access to AT is linked to better employment and earnings among people with disabilities. While AT has an enormous significance in the employment, the key is the provision of it by overcoming said barriers [233]. Breakbot, developed by [234], is an emotionally expressive companion robot designed to encourage employees to take regular breaks and engage in social interactions. As an assistive intervention, it contributes to positive workplace outcomes by improving work pacing, promoting wellbeing, and fostering social connectedness among employees. The motionEAP system, developed by [235], was designed to enhance workplace efficiency and support production processes through motion detection and projection-based feedback. The system uses a top-mounted projector to display real-time guidance on whether workers select the correct components and tools during assembly.

Looking at the Category column in Table 2.3, it starts to become more apparent and see how the contributing factors connect to the impacted ones, forming the basis for the derived metrics. Within the mental health theme in the education sector, categories such as financial wellbeing and curriculum act as contributing factors that directly influence performance. These represent underlying conditions or structures that shape how students experience and manage stress, ultimately affecting their academic outcomes.

In contrast, within the workplace setting, creators emerge as the primary contributing factors that influence outcomes, which encompass performance, productivity, and satisfaction. This distinction shows that while educational factors tend to originate from environmental and structural aspects, workplace factors are more closely tied to the technological and operational demands placed on individuals.

Together, these relationships highlight a clear flow from contributing elements to impacted outcomes, where the former explains the cause and the latter defines the measurable effect. The impacted categories, performance, satisfaction and productivity serve as the key outcome measures used to evaluate how different factors influence user wellbeing, experience, and technostress. These categories form the basis for the derived metrics presented later in the chapter, allowing for consistent measurement and comparison across both educational and workplace contexts.

Table 2.3: Taxonomy Categorised Citations

Theme	***Sector***	***Category***	***Paper***	***C***	***S***
Mental Health	Education	Performance*	[54], [55], [56], [65], [58], [59], [60], [61], [62], [63], [64], [65], [66], [69], [70], [71], [72], [73], [74]	19	***76***
		Financial**	[75], [84], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86]	12	
		Curriculum**	[95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114], [115], [116], [117]	23	
	Workplace	Creators**	[121], [122], [123], [124], [125], [126]	6	
		Interventions	[127], [128], [129], [130], [131], [132], [133], [134], [135], [136]	10	
		Outcomes*	[137], [138], [139], [140], [141], [142]	6	
Technostress	Education	Experience**	[146], [147], [148]	3	***41***
		Creators**	[149], [150], [151], [152], [153], [154], [155], [156], [157], [158], [159]	11	
		Outcomes*	[160], [161], [162], [163], [164], [165], [166]	7	
		Interventions	[167], [168]	2	
	Workplace	Experience**	[171], [172], [173]	3	
		Creators**	[174], [175], [176], [177]	4	
		Outcomes*	[178], [179], [180], [181]	4	
		Interventions	[182], [183], [184], [185], [186], [187], [188]	7	
Assistive Technology	Education	Inclusion**	[193], [194], [195], [196]	4	***42***
		Engagement**	[197], [198], [199], [200], [201], [202], [203]	7	
		Performance*	[204], [205], [206], [207], [208], [209]	6	
	Workplace	Inclusion**	[211], [212], [213], [214], [215]	5	
		Barriers**	[216], [217], [218], [219], [220], [221], [222], [223], [224], [225], [226], [227]	12	
		Outcomes*	[228], [229], [230], [231], [232], [233], [234], [235]	8	
Total					***159***

Key: ** - Impacted Factor, ** - Contributing Factor*

Figure 2.3 illustrates the hierarchical structure of the proposed taxonomy in Table 2.3, which organises the literature into three overarching themes: Mental Health, Technostress, and Assistive Technology. Each theme is further divided into educational and workplace contexts to capture the differences and overlaps between learning and professional environments. Within these contexts, multiple sub-categories were identified to represent both contributing and impacted factors, ultimately linking to key outcomes, namely performance, productivity, and satisfaction. These three particular sub-categories emphasise their importance as metrics for technostress.

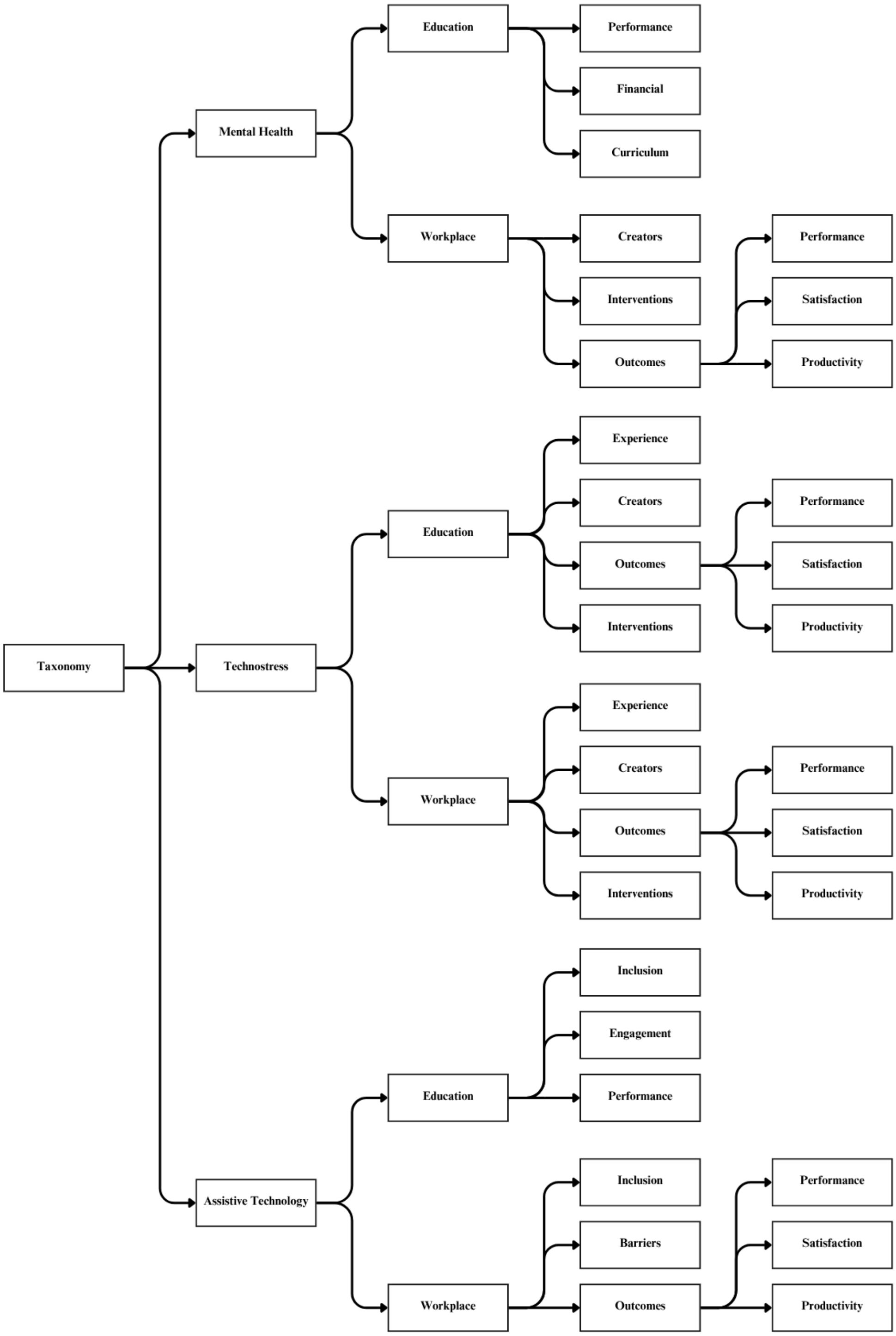


Figure 2.3: Taxonomy Branches

2.6 Discussion

2.6.1 Addressing Research Question One

***Question*:** What are the primary factors and outcome measures used in the existing literature to characterise technostress and its impact in STEM contexts? (RQ1)

***Findings*:** Stress and anxiety negatively impact students' performance, creating a knock-on effect on degree attainment. Financial wellbeing further shapes this relationship, as financial stress can exacerbate academic strain and reduce performance outcomes. Conversely, well-designed curricula can positively influence wellbeing when supported by appropriate tools and student-centred assessments that enable students to meet deadlines effectively. Within this context, techno-stressors emerge as key factors influencing performance, productivity, and satisfaction. Students' prior experience with technology also determines their comfort level and susceptibility to technostress, shaping how effectively they adapt to technological changes and adoption processes. AT serves as a critical enabler across these domains by enhancing accessibility, fostering inclusion, and supporting active participation. Its integration not only boosts engagement but also strengthens performance, bridging the gap between wellbeing, technology use, and academic achievement.

For developers, a similar pattern was observed, where the creators of mental strain produced a knock-on effect on key outcomes such as performance, productivity, and satisfaction. Experience emerged as a critical factor, shaping how effectively developers navigated these challenges. Those with greater professional exposure and familiarity with complex development tools were associated with higher resilience to stressors, whereas less experienced developers were more susceptible to overload and uncertainty when faced with demanding workflows or rapid technological change

***Discussion*:** Performance, productivity, and satisfaction (collectively referred to as outcomes) are the primary impacted factors that appear across the explored literature themes. They were therefore selected as the derived metrics for this research moving forward, providing a reliable and evidence-based means of assessing user experience, wellbeing, and technostress levels. Students' mental health strongly influences their outcomes in STEM. Technostress can further hinder these outcomes, while supportive digital tools and strategies that enhance accessibility and engagement can help mitigate these challenges and improve success.

2.6.2 Addressing Research Question Two

Question: How can existing research that intersects mental health, technostress, and assistive technology be systematically synthesised and categorised? Where are the key gaps in current research that a taxonomy can help identify? (RQ2)

Findings: Mental health is the most extensively studied domain, particularly in educational settings, reflecting the recognised importance of student wellbeing and its impact on academic outcomes. Assistive Technology is moderately explored, mainly in workplace contexts, with evidence suggesting it can improve accessibility, engagement, and performance. Technostress, however, remains the least examined theme ($n = 41$, n here refers to the number of papers in Table 4) amongst the rest . Although studies identify its creators and document their effects on outcomes, there is little research on strategies or interventions that address these stressors, despite the considerable attention given to technostress creators within the literature ($n = 11$).

Discussion: A taxonomy has been developed based on systematic review of the literature intersecting the themes, and re-summarising why the categorisation of the taxonomy as it stands can help in general in research that intersects those themes. The taxonomy can serve as an analytical tool for categorising future studies in these themes and/or research questions can be derived from specific nodes/branches of this taxonomy. The findings revealed that there remains a significant research gap concerning technostress within educational settings. While numerous studies have identified the creators of technostress and examined their effects on key outcomes such as performance, productivity, and satisfaction, there remains a lack of research in the education sector that has implemented and evaluated practical interventions to mitigate technostress while simultaneously improving student outcomes. This motivates the contributions in the upcoming chapters particularly around how the interventions can be embedded in educational tool design.

The taxonomy further highlights an important connection between AT and technostress mitigation. Across the assistive technology literature, considerable emphasis is placed on designing systems that adapt to user needs through mechanisms such as personalisation, user control, accessibility, and supportive feedback. Although these principles are often discussed in the context of accessibility, they are equally relevant to the design of educational technologies that aim to support cognitive and emotional wellbeing. Within IDE environments, such principles can be used to reduce unnecessary cognitive burden, improve usability, and provide more supportive user experiences. As a result, the AT domain offers both theoretical foundations and practical design strategies that can inform the development of technostress-aware IDEs capable of responding more effectively to the needs of student programmers.

Such absence suggests a significant opportunity to develop approaches that directly address technostress. This gap is particularly evident within STEM disciplines, especially in the field of Computing, where students engage extensively with technology as an integral part of their learning, coursework, and assessments. The continuous interaction with digital tools both within and beyond academic contexts can intensify technostress levels, making it essential to design and assess targeted strategies that support student wellbeing and academic success.

2.7 Threats to Validity

2.7.1 Internal

As the field continues to evolve, it is acknowledged that not every aspect of the literature could be fully captured, even with a systematic and structured approach. To minimise this potential limitation, the study employed a rigorous and transparent methodology. Clear inclusion and exclusion criteria were established to ensure both relevance and quality, while a snowballing technique was applied to identify additional high-impact studies that may not have appeared in the initial database search. However, snowballing may introduce bias towards frequently cited or highly visible publications, potentially underrepresenting less cited but still relevant studies.

Additionally, publication bias may have influenced the findings, as studies reporting positive or statistically significant outcomes are generally more likely to be published than studies reporting neutral or negative results. Language bias is also acknowledged, as only English-language publications were included in the review process, potentially excluding relevant international research. Furthermore, although major academic databases relevant to software engineering, HCI, and educational technology were searched, limitations in database coverage may have resulted in the omission of relevant grey literature, industry studies, or publications indexed outside the selected sources.

The screening, categorisation, and thematic synthesis processes were primarily conducted by a single researcher, which introduces the possibility of subjective interpretation. To reduce this threat, predefined selection criteria, iterative review procedures, and structured categorisation approaches were consistently applied throughout the review process. Despite this, the combination of systematic search procedures and adaptive review techniques strengthened the internal validity of the taxonomy by supporting a comprehensive and representative body of evidence.

2.7.2 External

The taxonomy is intentionally scoped to the educational and workplace contexts, where the constructs of mental health, technostress, and assistive technology most prominently intersect. As a result, it does not extend to every possible domain in which technostress or related psychosocial factors may emerge. To minimise this limitation, the focus was deliberately aligned with the research gap identified in Section 2.6.2, which highlights the need to better understand the educational setting in comparison to the workplace.

Accordingly, the taxonomy prioritises conceptual clarity and cross-contextual relevance between these two environments, rather than the detailed nuances of workplace dynamics themselves. While the findings provide a foundation for understanding technostress-related interventions and assistive technologies within educational and professional settings, caution should be exercised when generalising the taxonomy to other domains, populations, or cultural contexts not explicitly represented within the reviewed literature. This is particularly important given the potential influence of publication bias, language restrictions, and database limitations on the diversity of included studies. Nevertheless, by synthesising evidence across both educational and workplace environments, the taxonomy aims to provide broader applicability and support future adaptation to adjacent domains and emerging technological contexts.

2.8 Summary

By mapping how contributing factors lead to core outcomes, the taxonomy (addressing RQ2) establishes a conceptual foundation for identifying measurable variables (addressing RQ1), later operationalised as derived metrics that underpin both the framework and biometric analysis presented in subsequent chapters. Addressing the research gap outlined in Section 2.6. The next chapter introduces the Iterative IDE Enhancement Framework, which builds directly upon these derived metrics. Centred around performance, productivity, and satisfaction, these metrics are systematically embedded within the framework to evaluate and refine practical interventions aimed at reducing technostress and enhancing the student experience within IDEs. This progression bridges the conceptual insights of the taxonomy with their applied implementation, linking theoretical understanding to actionable design strategies for technostress-aware programming environments.

Chapter *Three*

3 Procedural Framework

3.1 Introduction

Building on the gaps identified in the preceding chapter (which identifies the key metrics of technostress being performance, satisfaction and productivity), this chapter advances the research by translating conceptual insights into a concrete design process, embedding the aforementioned metrics.

While Chapter Two established the theoretical foundations and defined key metrics linking technostress, wellbeing, and software interaction outcomes, the present chapter focuses on operationalising these metrics within a practical, feedback-driven framework for IDE enhancement. The framework is designed to support systematic evaluation and refinement of IDE features with the explicit goal of improving user experience and reducing technostress.

The chapter centres on the findings of Chapter Two's identified gap concerning technostress within educational settings, with computing students as the primary context of investigation. This population represents one of the most technology-intensive user groups, engaging continuously with IDEs, development tools, and digital platforms as part of their learning, assessment, and skill development. As such, they provide a critical lens through which the effects of IDE design decisions on technostress, performance, and productivity can be meaningfully examined.

Programming has regained considerable attention over the last decade [236] and is widely recognised as a challenging and demanding activity that can be described as a "*process of developing and implementing various sets of instructions to enable a computer to perform a certain task, solve problems, and provide human interactivity*" [237]. The main tool used for programming is an IDE, a software program that incorporates all tools required for a software development project, including an editor, compiler, and debugger, such tool aggregates several facets of software development in one package which greatly increases programmers' efficiency. IDEs are at the core of programming education and practice. IDEs serve as the primary workspace for coding activities and provide essential tools for writing, debugging, and testing code. For many students, an IDE is their first exposure to professional development tools, making it a crucial component of their learning experience. Whether self-taught or formally trained, all learners must learn to navigate these environments as part of their skill-building journey. Given the learning curve associated with IDEs, past research [238] has explored how friendly IDEs really are.

The contribution of this chapter is the development of an Iterative IDE Enhancement Framework, designed to mitigate technostress through a systematic and data-informed process of IDE modification and evaluation. Together, the derived metrics and the framework provide a dual contribution: the metrics (presented in chapter 2) enable the quantification of technostress and user experience factors, while the framework (presented in this chapter) operationalises these measures to iteratively refine the IDE. Through this process, it is possible to identify which modifications most positively or negatively influence specific technostress triggers and related user outcomes such as performance, satisfaction, and productivity.

In addition to presenting the framework's structure and theoretical foundation, this chapter provides evidence of its practical application through the implementation and evaluation of the modified IDE. By combining quantitative and qualitative insights, the framework serves as both an identification and an intervention tool, offering an approach with potential for identifying technostress triggers within an IDE and thereafter reducing technostress and improving user experience in educational programming environments.

3.1.1 Problem Statements and Research Questions

Despite growing recognition of technostress as a factor influencing technology use, it remains insufficiently addressed within IDE design research, particularly in educational contexts. Existing work has largely focused on identifying techno stressors and associated outcomes, yet there is limited guidance on how these insights can be embedded into a structured design process. Although Chapter 2 defined relevant metrics that connect technostress, user experience, and task outcomes, these metrics currently lack an integrated mechanism through which they can inform iterative software design decisions.

Development tools, particularly IDEs are shaped by factors like UX, which is overlooked by how technostress, which can directly impact UX thus ending up having a knock on effect on performance (efficiency of tool usage), satisfaction (perceived ease and enjoyment), and productivity (task completion effectiveness) [239]. Given that technostress can directly shape how users engage with development tools there opens up a clear need for a framework that embeds these metrics within a systematic, feedback-driven process. Addressing this gap is essential for informing the design of more supportive, technostress-aware programming environments that respond to the lived experiences of their end users

The objectives are to create a structured approach which aids IDE designers in incorporating the following user experience aspects when designing IDEs:

(Obj 1) Studying the demographic-specific impact of IDE modifications on technostress levels measured in terms of performance, productivity and satisfaction.

(Obj 2) Enhancing student developers' performance, productivity, and satisfaction as measures of technostress.

To achieve these objectives, the study addresses the following research questions with respect to any changes/modifications made in the design of an IDE:

(RQ 1) How do different genders perceive the changes in the IDE differently?

(RQ 2) How do different education levels perceive the changes in the IDE differently?

(RQ 3) How do novice and experienced student developers perceive the changes in the IDE differently?

(RQ 4) How do different ethnic groups perceive the changes in the IDE differently?

(RQ 5) How do tailored modifications to IDEs impact technostress – measured in terms of performance, productivity, and satisfaction?

(RQ 6) Which features of the modified IDE contribute to reducing technostress factors?

RQs 1-4 (Obj 1) examine whether the effects of tailored IDE modifications are equitably distributed across key sociodemographic groups that are known to experience technology differently (as argued in Section 3.2.2). RQs 5-6 (Obj 2) examine their effectiveness in relation to outcomes.

3.1.2 Contribution and Significance

These questions guide the investigation into how IDE modifications shape students' experiences, ultimately forming the core contribution of this chapter:

(C1) *Framework for technostress-centric IDE enhancement*: The study proposes a structured approach for iterative IDE modifications based on user feedback and data-driven refinements, serving as a framework for future research and development in the IDE design.

The novelties of the contribution are as follows:

(N1) *Capturing technostress as a major concern within IDE design*: By using systemically derived metrics (presented in Chapter 2) that are related to technostress as drivers for design decision-making, this framework is first (to the user's knowledge) to treat technostress as a primary design concern within IDE design in educational settings.

3.1.3 Structure

Section 3.2 outlines the conceptual background of the framework, detailing its theoretical foundations and the process through which it was derived. Section 3.3 reviews existing frameworks and critically positions them in relation to the proposed technostress-aware framework, highlighting areas of alignment and differentiation. Section 3.4 presents the framework, detailing its structured phases (Preparation and Design, Participant Selection and Validation, and Iterative Evaluation and Refinement) which can be applied iteratively by IDE designers. It outlines how the framework supports a technostress-centric approach to evaluating IDE modifications in relation to the research objectives above. Section 3.5 details the methodology used for the practical study illustrating the framework's usage. It covers the study design, including the selection of participants, the experimental setup, and the evaluation criteria with respect to the research objectives. This section also discusses the ethical considerations addressed to ensure responsible research practices, such as informed consent, data privacy, and participant well-being. Furthermore, it provides an overview of the IDEs used in the study, along with the specific modifications implemented to assess their impact on usability, technostress, and overall user experience. Section 3.6 presents the results obtained through evaluating the framework using the set up from the previous section. It illustrates how the framework addresses each of the research questions highlighting how modifications suggested by the framework lead to changes in usability, acceptance, performance, productivity, satisfaction, and technostress (in alignment with Obj 1 and 2). The findings are reported both at the overall cohort level and through demographic comparisons, analysing differences based on gender, education level, coding experience, and ethnicity. This breakdown allows for a more detailed understanding of how different user groups interact with the modified IDE and whether the changes benefit users equitably. Section 3.7 systematically maps each modification to its corresponding technostress dimension (techno-overload, techno-complexity, techno-invasion, techno-insecurity, or techno-uncertainty) and identifies the intended user outcomes (including enhancements in performance, productivity, or satisfaction). To clearly illustrate these relationships, presenting a comprehensive mapping table that indicates how each modification predominantly links to specific technostress triggers, either contributing to their reduction or, in some cases, inadvertently increasing them. Also highlighting the corresponding outcomes affected by each modification, this offers a consolidated view of the overall impact. Section 3.8 discusses the threats to validity associated with the framework, acknowledging methodological and conceptual limitations. Section 3.9 summarises the findings, linking them back to the chapters objective RQs.

3.2 Background

The interplay between software design and user well-being has gained increasing attention in recent years, particularly in the context of IDEs used by novice and student developers. Traditional IDEs have largely prioritised functionality and extensibility over user experience and emotional well-being, often overlooking the cognitive and emotional demands they place on users [240], [241].

A major factor influencing developers experiences is technostress, which refers to the strain individuals experience due to the use of technology. In the context of IDEs, technostress can arise from various triggers as described in Table 3.1. Despite growing awareness of the psychological and emotional components of IDE interaction, to this day no studies proposing frameworks that explicitly connect IDE design features to the mitigation of technostress in student populations despite the presence of work studying technostress in workplace settings [242], [243], [244].

Table 3.1: Technostress Terms in the Context of IDE Usage Adapted from [9]

Stressor	***Definition***
Techno-overload	Developers feel overwhelmed by numerous features, complex toolbars, and extensive plugin options, leading to cognitive overload.
Techno-complexity	IDEs contain intricate settings, hidden functionalities, and steep learning curves that make it challenging for users to navigate efficiently.
Techno-invasion	Persistent notifications, autosuggestions, and debugging alerts interrupt workflow, reducing focus and deep work.
Techno-insecurity	Developers worry about keeping up with frequent IDE updates, new frameworks, or evolving programming standards.
Techno-uncertainty	Unpredictable IDE updates, inconsistent UI modifications, or deprecated features disrupt users' established workflows.

3.2.1 Conceptual Origins and Theoretical Grounding

The framework is conceptually grounded in three complementary research strands: (1) technostress theory and stressor–strain models, (2) HCI evaluation practices, and (3) iterative software design methodologies. From technostress research, the framework adopts the premise that stress arises from specific technological characteristics, such as complexity, overload, and uncertainty, and that these stressors manifest as emotional and behavioural responses during interaction. From HCI, it draws on mixed-method evaluation principles, recognising the value of combining subjective feedback with behavioural and physiological measures. From software engineering, it adopts an iterative, feedback-driven design philosophy in which empirical user data informs incremental system refinement.

3.2.1.1 Design Principles and Their Role in Framework Grounding

The proposed framework is strongly grounded in foundational HCI theories that conceptualise interaction as a cognitive, goal-driven process shaped by system design choices. In particular, it draws on Shneiderman's Eight Golden Rules [245], Norman's Theory of Action [246], and complementary HCI evaluation frameworks that emphasise usability, cognitive load, and iterative refinement. These theories collectively provide the methodological and theoretical justification for the framework's three-stage structure.

Shneiderman's Eight Golden Rules of Interface Design articulate core principles for reducing user error, cognitive overload, and interaction friction through consistency, informative feedback, error prevention, and user control. These rules are particularly relevant in the context of IDEs, where users engage in prolonged, cognitively demanding interactions involving frequent context switching, error resolution, and system feedback interpretation.

Norman's Theory of Action conceptualises interaction as a cyclic process involving goal formation, execution, system response, and evaluation. Central to this model are the gulf of execution (the gap between a user's goal and the physical actions needed to operate a system) and gulf of evaluation (the gap between the system's state and the user's interpretation of that state), terms invented in 1986 by [247], which describe mismatches between user intentions and system affordances, and between system feedback and user interpretation, respectively. These gulfs are particularly salient in complex software systems such as IDEs, where abstract operations, dense interfaces, and delayed feedback can amplify cognitive strain.

In addition to Shneiderman and Norman, the framework draws on broader HCI experimental and evaluation traditions, including mixed-method usability evaluation [248], task-based interaction analysis, and ecologically valid experimental design. These approaches emphasise the importance of aligning tasks with real-world usage patterns while maintaining experimental control, an essential requirement when integrating behavioural, subjective, and physiological data

At the framework level, these HCI traditions reinforce the integration of iterative software design methodologies, ensuring that evaluation is not treated as a terminal phase but as an ongoing mechanism for design refinement. This directly supports the framework's positioning within technostress research, where reducing stress requires not only measurement but design-informed intervention.

Collectively, these HCI theories operationalise the framework's conceptual grounding. Shneiderman and Norman provide the interaction-level mechanisms through which technostress creators manifest during use, HCI evaluation practices justify the staged methodological structure, and iterative design principles explain why evaluation outcomes are reintegrated into system redesign. As a result, the framework is not a generic usability model but a technostress-aware HCI framework, explicitly designed to identify, measure, and mitigate stress-inducing characteristics of IDEs through theory-driven experimental design and iterative refinement.

3.2.2 Exploring User Sociodemographic

3.2.2.1 Gender Comparison

The decision to analyse gender differences in the study stems from the need to understand how diverse user groups interact with the IDEs when programming. Studies suggest that gender-based variations exist in the perceived usability, efficiency, and accessibility of programming interfaces [249], [250], [251]. Research indicates that males and females differ in perceived usability, efficiency, and accessibility of programming interfaces, with women prioritising usability and inclusivity and men focusing on performance and task completion [252], [253], [254], [255], [256]. Women's broader evaluation criteria lead to more varied responses, emphasising aesthetics and inclusivity [250], [256], while men's efficiency-focused approach results in more consistent feedback [257], [258]. Understanding these differences aids in assessing the impact of IDE modifications on diverse users.

3.2.2.2 Education Comparison

Analysing education level differences in this study stems from the need to assess how individuals with varying levels of education interact with IDEs during programming tasks. Studies suggest that individuals with higher education levels are more proficient in handling complex software environments, while those with less formal education may encounter challenges when navigating advanced IDE features [249], [250]. Understanding these variations is crucial in evaluating whether the modifications introduced cater to users from diverse educational backgrounds, ensuring equitable learning experiences and usability for all [251].

3.2.2.3 Experience Comparison

Examining differences based on coding experience is essential for understanding how novice and experienced programmers interact with IDEs. Prior research suggests that users with varying levels of coding expertise exhibit different behaviours when navigating development environments. Experienced programmers tend to utilise advanced IDE features more efficiently, leveraging shortcuts, debugging tools, and automation functions, whereas novice users often struggle with complex functionalities and rely more on basic features and external resources [252]. Additionally, coding experience influences problem-solving strategies and error-handling approaches. Studies indicate that experienced developers were associated with higher efficiency in debugging and code optimisation, while less experienced users may require additional support, such as guided error messages, intelligent code suggestions, and simplified navigation [253]. By incorporating coding experience level as a factor in the analysis, this study assesses whether the modifications provide usability improvements for all users, regardless of their expertise levels [254]. This approach helps identify whether the changes lead to enhancements for beginners while maintaining efficiency and flexibility for those with more experience [255]. While there may be some overlap between experience and education, it is important to distinguish that experience often encompasses self-directed learning and practical application, whereas education covers students being taught computing concepts, which may or may not necessarily translate into self-learning.

3.2.2.4 Ethnicity Comparison

Analysing ethnic differences in the study is crucial to understand how diverse user groups interact with the IDE. The software development industry has recognised the importance of diversity and inclusion, yet disparities persist in representation and experiences among different ethnic groups. A systematic literature review on perceived diversity in software engineering revealed that only 2% of studies focused on racial diversity, indicating a significant gap in understanding how ethnicity influences software development practices and tool usage [256]. Furthermore, research highlights that the design of software tools may not always account for the diverse cognitive and technical backgrounds of users, leading to differences in perceived usability and efficiency across ethnic groups [257]. Research has also shown that ethnic diversity within software development can influence productivity and collaboration dynamics [258]. By including ethnicity comparison in the analysis, this study ensures that the modifications introduced are evaluated with a broader perspective. This approach helps determine whether the changes are equitable and beneficial for all users, mitigating potential disparities.

3.3 Related Work

While much research has focused on identification and measurement of stressors, a smaller but growing body of work addresses frameworks for mitigating technostress. A major contribution in this space is the formation and mitigation model by [259], which unpacks how technostress forms and can be mitigated in personal IT use, introducing concepts such as IT affordances and actualisation costs that shape stress formation and user coping processes. This work complements organisational and individual strategies by illustrating behavioural mechanisms through which stress can be both generated and alleviated. More recently, [260] proposed a seven-step organisational framework for business and IT leaders aimed at systematic reduction of technostress across departments, based on a case study that implemented targeted measures and linked them to outcomes such as reduced sick days and improved performance. Such frameworks represent a shift toward practical, system-level intervention design, but tend to focus on organisational policy and structural change rather than in-task design adjustments at the human–technology interface. The work by [261] synthesises workplace technostress management strategies, highlighting interventions such as training, policy design, and organisational support structures.

Other complementary approaches include work by researchers such as [262], which examines coping with technostress-related goal hindrances in personal technology use and highlights goal prioritisation as part of mitigation strategy, stressing the interplay between user goals and stress responses. Additionally, [263] applies transaction stress models to digital workplaces to exhibit how preventive measures grounded in theory can reduce strain responses directly, offering frameworks that span primary, secondary, and tertiary prevention strategies.

3.3.1 Positioning Relative to Existing Framework

As shown in Table 3.2, existing technostress frameworks provide important theoretical, methodological, and practical foundations for understanding the causes, effects, and mitigation of technostress. However, these frameworks predominantly operate at the organisational, behavioural, or conceptual level, rather than embedding technostress-awareness directly within the software design process.

The comparison is not intended to rank frameworks, but rather to position the proposed framework relative to existing approaches within the context of its specific design goals. While existing frameworks provide important theoretical, methodological, and practical contributions to technostress research, none explicitly integrate technostress theory, mixed-method evaluation, and iterative IDE refinement within a single framework. In contrast, the proposed framework translates established technostress constructs into actionable, interface-level interventions by integrating user feedback and empirical evidence into iterative design decisions. This positions the framework as a complementary extension to existing technostress research, specifically tailored to programming environments.

Table 3.2: Comparison of Existing Related Work and the Proposed Framework

Framework /Study	***Primary Context***	***Stress Focus***	***Data & Measures***	***Temporal Resolution***	***Design Integration***	***Relevance to Proposed Framework***	***Key Limitations Addressed by This Work***	***N1***
[259]	Personal IT use	Stress formation & mitigation	Surveys, interviews	Longitudinal / reflective	Behavioural coping strategies	Demonstrates mitigation mechanisms and coping responses that complement system-level design interventions.	Focuses on user coping, not system or IDE redesign	
[260]	Organisational / enterprise IT	Organisation-wide technostress	Surveys, HR metrics	Longitudinal	Policy and management-level interventions	Illustrates structured technostress management approaches and reinforces the need for intervention at multiple levels.	Not suitable for fine-grained, in-task software design refinement	
[261]	Workplace IT	Technostress management	Systematic literature review	N/A	Descriptive synthesis	Consolidates existing mitigation strategies and identifies opportunities for translating technostress knowledge into practical processes.	Identifies strategies but does not operationalise them within design processes	
[262]	Personal technology use	Goal hindrances and coping mechanisms linked to technostress	Qualitative, multidimensional goal analysis	Hierarchical and dynamic coping perspective	Limited direct integration into software design processes.	Highlights how technostress emerges through conflicts between user goals and technology use, providing insight into user-centred mitigation considerations.	Does not provide actionable IDE-level design pathways or empirical physiological grounding	
[263]	Personal IT use	Formation and mitigation processes of technostress	Qualitative interviews, affordance-based analysis	Process-oriented but retrospective	Conceptual design implications	Demonstrates how specific technological affordances can contribute to or mitigate technostress, informing design-oriented intervention strategies.	Lacks structured translation into implementable design modifications	
Proposed TIDEs Framework	IDE-based programming (education)	IDE-induced technostress	Task outcomes, surveys (Quantitative and Qualitative)	Post-task	Iterative, feedback-driven IDE refinement	Extends prior work by combining technostress theory, mixed-method measurement, and iterative design refinement within a programming-specific context.	Bridges theory, emotion-aware measurement, and actionable IDE design decisions	*

3.4 Presenting the Framework

3.4.1 Description

The framework process presented in Figure 3.1, provides a structured and iterative process to assess and optimise for technostress-awareness in IDE design. The framework is split into three key stages: (1) Preparation and Design, (2) Participant Selection and Validation and (3) Iterative Evaluation and Refinement, which all will be discussed in detail in the rest of this section. Each stage is designed to ensure reliability and adaptability, enabling a comprehensive evaluation of modifications based on user feedback and performance, satisfaction, and productivity outcomes as measures of technostress. This subsection is intended to guide IDE designers who are the primary users as the implementers of the framework, whilst the target IDE users (i.e. students) will be integral to applying the stages of the framework.

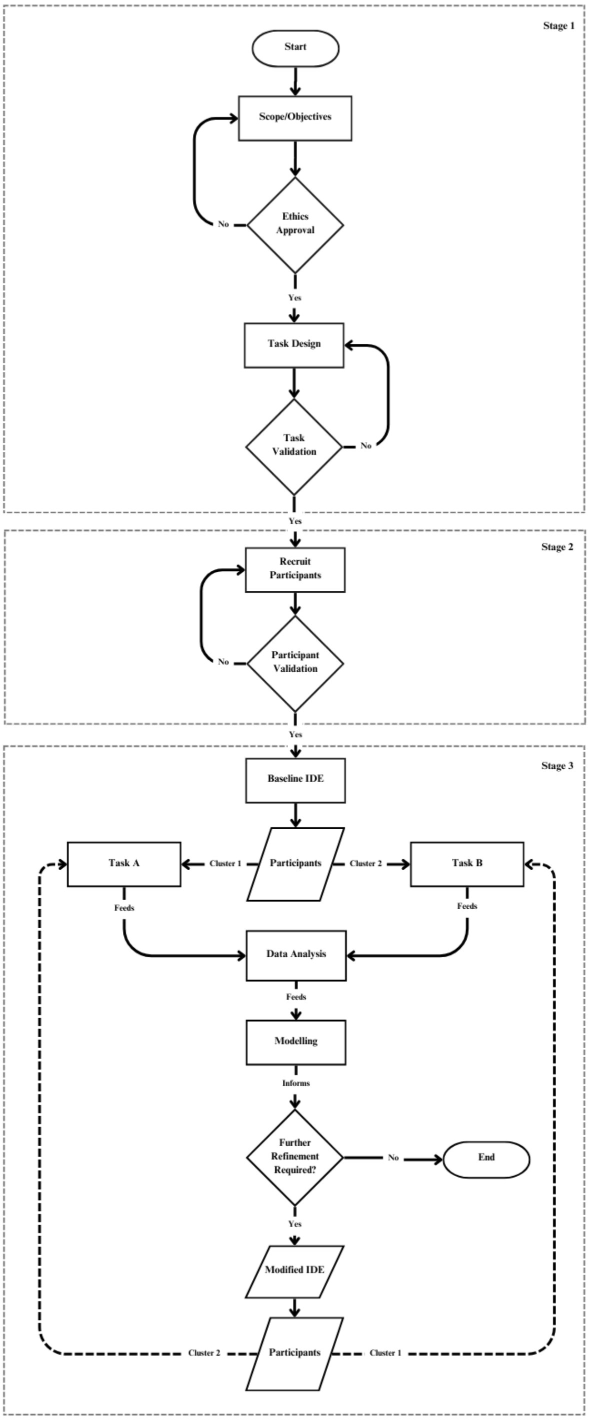


Figure 3.1: Iterative IDE Enhancement Framework

3.4.2 Stage 1: Preparation and Design

Grounded in HCI evaluation practice and iterative design methodology, this stage establishes the structural conditions under which technostress can be meaningfully examined. It defines the IDE context and evaluation scope, ensures ethical compliance, and operationalises programming tasks that reflect realistic development activities. Task design and validation draw on HCI principles to ensure cognitive demands are appropriate to participant expertise, supporting reliable measurement of interaction outcomes. By stabilising the experimental environment and task structure, this stage provides the methodological foundation for subsequent participant-centred evaluation and technostress analysis.

3.4.2.1 Scope/Objectives

The first step involves defining a specific IDE to which the framework will be applied and which components within it will be targeted by the framework. Importantly, the framework is designed to be implemented and guided by IDE designers or software developers, who are responsible for executing each phase of the process.

The overarching objective is to assess how adaptive changes to the IDE can improve user experience and reduce technostress. Importantly, even if an alternate or secondary IDE is used within the experiment, the evaluation remains consistent in measuring the afore mentioned core factors. This ensures the framework's principles and objectives are preserved regardless of the IDE platform, allowing for flexibility in implementation while maintaining the integrity of the evaluation criteria. This scoped approach enables the development of a repeatable framework that integrates user feedback, and iterative design, ultimately supporting more mentally supportive development environments for programmers.

3.4.2.2 Ethics Approval

To uphold ethical integrity, approval is to be obtained to ensure participant safety, confidentiality, and compliance with institutional and legal standards. This step is particularly crucial for studies involving human subjects, safeguarding both participants and researchers while demonstrating a commitment to responsible research practices.

3.4.2.3 Task Validation

Finally comes the programming/coding tasks or exercises that are required to be designed to not only align with the study’s goals but also encourages use of the IDEs key areas as defined within scope. These tasks should be carefully designed to engage participants in meaningful ways, eliciting behaviours and responses that provide valuable insights. Well-structured tasks enhance data accuracy and depth, ensuring they reflect realistic usage patterns. Before implementation, the designed tasks must undergo validation to confirm their appropriateness for the participants' expected skill levels and knowledge. This ensures that the tasks are relevant, manageable, and conducive to high-quality data collection. Validation enhances the reliability of results by guaranteeing that the exercises are clear, achievable, and aligned with the study’s purpose. This validation step supports the internal validity of subsequent comparisons between baseline and modified IDE conditions.

3.4.3 Stage 2: Participant Selection and Validation

Also informed by experimental HCI methodology, this stage focuses on ensuring an appropriate and reliable study population rather than task design. Computing students are selected as a high-exposure user group whose routine interaction with IDEs makes them particularly sensitive to system-induced stressors. Validation procedures are applied to ensure demographic relevance, task familiarity, and methodological consistency, mitigating threats to construct and internal validity commonly reported in technostress studies.

3.4.3.1 Participant Selection

Selection should focus on selecting individuals who meet predefined criteria, ensuring their background aligns with the scope’s requirements. By targeting participants from representative current or prospective demographics, this step enhances consistency and interpretability in the results, allowing for meaningful analysis and valid inferences from the collected data. Deliberate representative sample is important for addressing RQ1-4.

3.4.3.2 Participant Validation

The validation process is to be conducted to confirm whether participants meet the inclusion criteria. This ensures that all participants have the necessary baseline qualifications, reducing variability due to differences in knowledge or experience levels. By maintaining consistency across participant groups, this step strengthens the reliability and accuracy of the study’s findings.

3.4.4 Stage 3: Iterative Evaluation and Refinement

This stage constitutes the primary contribution of the framework and is explicitly grounded in technostress theory and iterative software design practices. It integrates post-task surveys (Chapter Three) with task outcomes to capture reflective indicators of technostress, which are then fed back into successive IDE design iterations. Rather than treating technostress as a purely latent or retrospective construct, this stage operationalises it through theory-driven metrics linked to performance, productivity, satisfaction, and emotional responses. Design decisions are therefore informed by established technostress creators, ensuring that refinement targets stress-inducing IDE characteristics rather than general usability issues.

3.4.4.1 Baseline IDE

Participants will first engage with the chosen IDE (selected by the developer/researcher) in its standard, unmodified form (Cycle 1), establishing a baseline for comparison. This control condition provides a reference point for evaluating modifications introduced in later cycles (Cycle 2 and beyond), ensuring that changes can be assessed for their impact on performance, productivity and user satisfaction.

3.4.4.2 Clustering

To control for individual differences and task-specific factors, through randomisation participants are to be divided into groups, Group A and Group B, each assigned different tasks. This structured grouping supports balanced, comparative analyses, allowing for more accurate conclusions about the effects of IDE modifications.

3.4.4.3 Data Analysis

In Cycle 1, participants' qualitative and quantitative feedback (primary data) should be gathered based on their interaction with the unmodified baseline IDE. This data provides insights into usability, acceptance and areas for improvement. Figure 3.2 outlines the data analysis and synthesis process which should be conducted in parallel using both qualitative and quantitative methods.

Given the affective and cognitive dimensions of technostress, reliance solely on quantitative usability scales or performance metrics, an approach still common in IDE evaluations and typically conducted post-task [264], [265], [266], is insufficient to capture the full complexity of user experience. To address this limitation, the framework adopts a mixed-method analytical strategy that integrates quantitative measures (e.g., usability scales and task outcomes) with qualitative feedback to enhance interpretive validity. Combining these data sources enables deeper insights than either method can provide independently [267], allowing recurring patterns and explanatory themes to emerge. Through triangulation, the analysis develops a more comprehensive and contextually grounded understanding of technostress dynamics within IDE interaction, ensuring that conclusions are empirically substantiated and theoretically coherent rather than a simple aggregation of isolated findings [268], [269]. The coded solutions (secondary data) that get submitted by participants are required to be analysed across cycles to assess how many task requirements are met and how many test cases are successfully passed. Each task should have a predefined set of requirements that participants are required to implement in their code, along with a set of test cases used to evaluate their solutions. This evaluation provides an objective measure of improvement in coding and the effectiveness of the modifications introduced in Cycle 2.

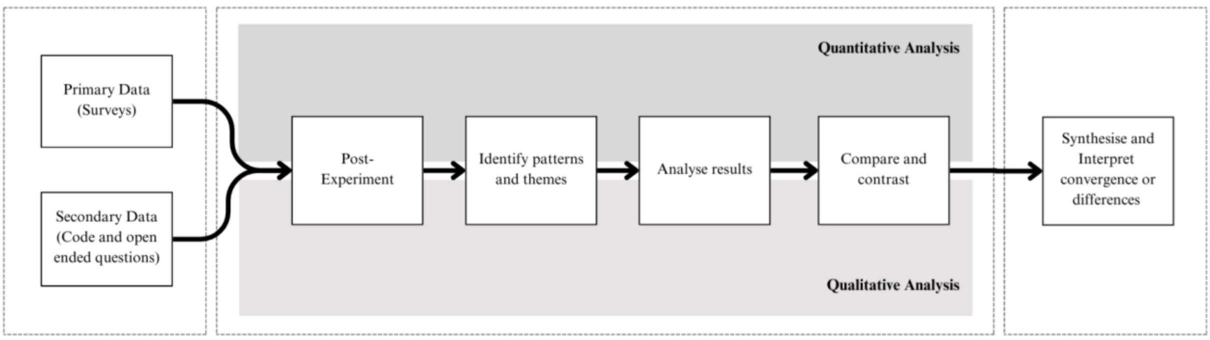


Figure 3.2: Analysis Approaches

The instruments used in this study serve different analytical purposes. SUS and TAM were employed to assess usability and technology acceptance respectively and should therefore be interpreted as indirect indicators of outcomes that may be influenced by technostress. In contrast, the SMI instrument and the technostress-trigger mappings were designed to more directly capture participants' perceptions of stress-related factors associated with IDE interaction. Accordingly, evidence relating to technostress is derived from the combined interpretation of SMI responses, qualitative feedback, and identified technostress triggers, rather than from SUS or TAM scores alone.

3.4.4.3.1 Quantitative Methods

3.4.4.3.1.1 System Usability Scale

Participants rated their perceptions of usability on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree) using System Usability Scale (SUS) by [270] however a revised version (Table 3.3). The revised version uses the same worded question with the addition of a placeholder (where the * is) to add an additional word to add further clarity in terms of whether the question being asked is in relation to the baseline or the modified version.

Responses are required to be analysed to evaluate participants' perceptions of usability of the baseline and modified system, with results broken down by Gender, Education Level, Coding Experience, Ethnicity, and Cluster Assignments. Positive SUS statements are grouped under "Agree/Strongly Agree," and negative statements under "Disagree/Strongly Disagree."

Table 3.3: System Usability Scale

		1	2	3	4	5
S1	I think that I would like to use this * system frequently					
S2	I found the * system unnecessarily complex					
S3	I thought the * system was easy to use					
S4	I would need the support of a technical person to be able to use this * system					
S5	I found the various functions in this * system were well integrated					
S6	I thought there was too much inconsistency in this * system					
S7	I would imagine that most people would learn to use this * system very quickly					
S8	I found the * system cumbersome to use					
S9	I felt very confident using the * system					
S10	I needed to learn a lot of things before I could get going with the * system					

Scale: 1-Strongly Disagree, 2-Disagree, 3-Neutral, 4-Agree, 5-Strongly Agree
** - placeholder for baseline/modified*

SUS Scores (SS) highlight trends and demographic variations. To derive the SS (Equation 3.1) formula (c) combines the x (a) and y (b) values to get the SS, which indicates the system's perceived usability. The SS ranges from 0 to 100. A perfect score of 100 reflects outstanding usability and an exceptional user experience. Scores of 70 or higher typically indicate good usability. Scores below 50 are considered poor, highlighting significant usability issues and the need for major improvements to boost user satisfaction.

$$\text{(a)}\ \ x = \sum OddStatements - 5$$
$$\text{(b)}\ \ y = 25 - \sum Even\ Statements$$
$$\text{(c)}\ \ SUS\ Score = x + y \cdot (25)$$

Equation 3.1: SUS Score Formula

3.4.4.3.1.2 Technology Acceptance Model

As for assessing participants' perceptions Technology Acceptance Model (TAM) [271] was used, however, a revised version (Table 3.4). Once again, the revised version uses the same worded question with the addition of a placeholder to add an additional word (where the * is) to add further clarity in terms of whether the question being asked is in relation to the baseline or the modified version.

TAM survey utilises a 7-point Likert scale (1 = strongly disagree, 7 = strongly agree). Responses are analysed to evaluate participants' acceptance of the baseline and modified systems thereby complementing the SUS analysis. Aggregated responses measured positive acceptance levels. These metrics provide insights into participants' Perceived Usefulness (PU) and Perceived Ease of Use (PEOU). PU refers to the extent to which a user believes that utilising a specific system will improve their job performance. PEOU refers to the extent to which a user believes that using a particular system will require minimal effort.

Table 3.4: Technology Acceptance Model Survey

		1	2	3	4	5	6	7
Perceived Usefulness								
PU1	Using * VSCode in my job would enable me to accomplish tasks more quickly							
PU2	Using * VSCode would improve my work performance							
PU3	Using * VSCode in my work would increase my productivity							
PU4	Using * VSCode would enhance my effectiveness							
PU5	Using * VSCode would make it easier to do my work							
PU6	I would find * VSCode useful in my work.							
Perceived Ease of Use								
PEOU7	Learning to operate * VSCode would be easy for me							
PEOU8	I would find it easy to get * VSCode to do what I want it to do							
PEOU9	My interaction with * VSCode would be clear and understandable							
PEOU10	I would find* VSCode to be flexible to interact with							
PEOU11	It would be easy for me to become skilful at using * VSCode							
PEOU12	I would find * VSCode easy to use							

*Scale: 1-Extremely Disagree, 2-Quite Disagree, 3-Slightly Disagree, 4-Neutral, 5-Slightly Agree, 6-Quite Agree, 7-Extremely Agree, * - placeholder for baseline/modified*

3.4.4.3.1.3 Usability Measures and Technostress Connection

Although technostress is often conceptualised as a psychological strain, as prior research in Chapter Two indicates, it manifests through interaction outcomes, particularly reduced performance, lowered satisfaction, and impaired productivity. In this framework, usability measures are therefore not treated as separate from technostress but as observable outcome indicators of underlying techno-stressors. System characteristics associated with technostress, are expected to negatively influence perceived usability and technology acceptance, which in turn affect task performance and user satisfaction.

Accordingly, instruments such as SUS and TAM are employed to capture these outcome-level effects, providing indirect but empirically grounded evidence of technostress impact. Rather than measuring stress perceptions alone, the framework operationalises technostress through its functional consequences during IDE interaction, aligning with stressor–strain models in technostress research. This enables systematic comparison across IDE configurations while avoiding overreliance on retrospective self-reports.

Within this framework, usability metrics, demographic factors, and qualitative insights collectively function as complementary lenses through which technostress is examined. This positioning ensures that the evaluation remains explicitly technostress-oriented, while still leveraging established HCI measurement practices to support rigorous and interpretable analysis.

3.4.4.3.1.4 System Modification Impact

In addition to the SUS and TAM a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree) System Modification Impact (SMI) survey (Table 3.5) was introduced to evaluate how the modifications impacted user performance, productivity and improved satisfaction while assessing reduction in technostress triggers. This survey is designed specifically for use within this framework, thereby reinforcing its technostress-centric orientation. It is introduced from Cycle 2 onwards, following the implementation of IDE modifications, in order to evaluate the impact of the proposed enhancements relative to the baseline condition established in Cycle 1. O1-O3 are directly operationalising the metrics presented in Chapter 2 and that TS1-TS5 are directly operationalising the established factors of technostress presented in Chapter 2 (section 2.3.1.2) and this Chapter (section 3.1.1).

Table 3.5: System Modification Impact Survey

		1	2	3	4	5
Outcome						
O1	The modified IDE Increased my Performance					
O2	The modified IDE Increased my Productivity					
O3	The modified IDE Increased my Satisfaction					
Technostress						
TS1	The modified IDE Decreased Techno-Overload					
TS2	The modified IDE Decreased Techno-Complexity					
TS3	The modified IDE Decreased Techno-Insecurity					
TS4	The modified IDE Decreased Techno-Invasion					
TS5	The modified IDE Decreased Techno-Uncertainty					
Scale: 1-Strongly Disagree, 2-Disagree, 3-Neutral, 4-Agree, 5-Strongly Agree						

Additionally, a follow up multiple-choice question (Table 3.6) was included to determine participants' preferred version of the system.

Table 3.6: System Modification Impact Follow Up

Preference	
P1	Which version of the IDE do you prefer? - Baseline VSCode (Cycle 1) - Modified VSCode (Cycle 2)

3.4.4.3.1.5 Task Assessment

A Task Assessment (TA) survey (Table 3.7) was administered from Cycle 1 onwards to evaluate perceived task difficulty and clarity. Participants rated task difficulty (TA1) on a 5-point scale (1 = easy, 5 = hard) and were additionally asked whether the task made sense within each cycle (TA2). Introducing the TA from Cycle 1 was essential to establish a baseline measure of task-related perceptions prior to any IDE modifications, ensuring that subsequent comparisons could distinguish between changes attributable to system enhancements and variations arising from task interpretation or inherent task complexity.

Table 3.7: Task Assessment Survey

TA1	Did the given task make sense? - Yes - No					
		1	2	3	4	5
TA2	How difficult did you find the given task?					

3.4.4.3.1.6 Scale Interpretation

As for interpreting the mean values for the five-point, a Likert scales (Table 3.8) was used in both the SUS and the SMI surveys. The Likert scale ranges from 1 to 5, where 1 represents Strongly disagree and 5 represents Strongly agree. The corresponding mean score intervals are divided into five distinct categories: 1.00–1.80 (Strongly disagree), 1.81–2.60 (Disagree), 2.61–3.40 (Neutral/Uncertain), 3.41–4.20 (Agree), and 4.21–5.00 (Strongly agree) [272]. These ranges allow for a consistent interpretation of participant responses and facilitate comparative analysis across both the SUS and SMI. For example, if a statement receives an average score of 2.35, it will fall within the Disagree category, indicating a generally negative response from participants.

Table 3.8: Five-point Likert Scale Mean Values and their Interpretation

Likert-Scale Description	***Likert-Scale***	***Likert Scale Interval***
Strongly disagree	1	1.00 - 1.80
Disagree	2	1.81 - 2.60
Neutral/Uncertain	3	2.61 - 3.40
Agree	4	3.41 - 4.20
Strongly agree	5	4.21 - 5.00

Table 3.9 provides the interpretation of mean values for the seven-point Likert scale used in the TAM. This scale ranges from 1 (Extremely disagree) to 7 (Extremely agree), offering a more nuanced understanding of participants' perceptions. The scale intervals are defined as follows: 1.00–1.85 (Extremely disagree), 1.86–2.71 (Quite disagree), 2.72–3.57 (Slightly disagree), 3.58–4.43 (Neutral/Uncertain), 4.44–5.29 (Slightly agree), 5.30–6.15 (Quite agree), and 6.16–7.00 (Extremely agree) [273]. This seven-point structure enables greater sensitivity in assessing participants' attitudes toward usability and acceptance compared to the five-point format. For example, if a statement receives an average score of 6.5, it will fall within the Extremely Agree category, indicating a generally a very positive response from participants.

Table 3.9: Seven-point Likert Scale Mean Values and their Interpretation

Likert-Scale Description	***Likert-Scale***	***Likert Scale Interval***
Extremely disagree	1	1.00 - 1.85
Quite disagree	2	1.86 - 2.71
Slightly disagree	3	2.72 - 3.57
Neutral/Uncertain	4	3.58 - 4.43
Slightly agree	5	4.44 - 5.29
Quite agree	6	5.30 – 6.15
Extremely agree	7	6.16 – 7.00

Together, these scales provide a consistent and transparent framework for interpreting participant responses. Scores above the agreement threshold in either scale indicate positive perception, forming a clear basis for identifying areas of success and opportunities for improvement within the study.

3.4.4.3.1.7 Statistical Measures

Data collected through the SUS, TAM, and SMI survey measures were analysed using descriptive and inferential statistics. To achieve this, a combination of descriptive statistics and inferential tests was applied, enabling a clear and transparent interpretation of survey outcomes across the baseline and modified IDE conditions.

3.4.4.3.1.7.1 Mean, SD and SEM

For each construct, the Mean (M) was calculated (Equation 3.2) to represent the central tendency of participant responses:

$$M = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Equation 3.2: Calculating Mean Formula

Where x_i denotes an individual participant's score and n is the sample size. To describe inter-participant variability, the Standard Deviation (SD) was computed as (Equation 3.3).

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - M)^2}$$

Equation 3.3: Calculating Standard Deviation Formula

SD was used to capture the extent to which individual responses varied around the mean. To quantify the precision of the estimated mean, the Standard Error of the Mean (SEM) was calculated (Equation 3.4).

$$SEM = \frac{SD}{\sqrt{n}}$$

Equation 3.4: Calculating Standard Error Mean Formula

SEM reflects the uncertainty associated with each mean estimate and was used to support interpretation of the reliability of group-level effects, particularly given the moderate sample size. Where appropriate, SEM was used to derive confidence intervals around the mean.

3.4.4.3.1.7.2 Reliability and Validation

To evaluate the internal reliability of the SUS, TAM, and SMI instruments, Cronbach's alpha (CA) was calculated for each construct across both experimental cycles. CA, introduced by Cronbach [274], is a measure of internal consistency used to assess the reliability of a psychometric instrument. Its value ranges from 0 to 1, with higher values indicating that the items measure the same underlying construct. Conversely, a lower value (closer to 0) suggests that some or all items may not be assessing the same dimension [275], this value can be calculated using the CA formula (Equation 3.5).

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k}\sigma_y^2}{\sigma_x^2}\right)$$

Equation 3.5: Cronbach's Alpha Formula

A High Internal Reliability (HIR) suggests that participants responded consistently to the survey items intended to measure each construct, supporting the validity of our inferences about perceived performance, productivity, and satisfaction in relation to the IDE modifications. HIR supports the validity of conclusions drawn about the IDE's effectiveness, particularly in enhancing performance, productivity, and satisfaction. It reflects that users not only perceived the modifications as intended but also experienced the interface in ways that aligned with the targeted improvements. On the other hand, a Low Internal Reliability (LIR) may indicate inconsistency in how participants perceived or responded to the modified IDE. This could suggest that the changes introduced were either unclear, inconsistently experienced, or differently interpreted depending on the user. In such cases, it becomes challenging to draw confident conclusions about the impact of the modifications because the measurement tools are not reliably capturing a shared experience

3.4.4.3.1.7.3 Paired Sample T-Test

Differences between the baseline (Cycle 1) and modified (Cycle 2) IDE conditions were evaluated using paired-samples t-tests. This test evaluates whether the mean of the differences between paired observations (i.e., the same participant across two cycles) is significantly different from zero. A paired approach is appropriate given the within-subjects design, where each participant contributes data to both Cycle 1 and Cycle 2.

First, the mean ($\bar{X}$) and standard deviation (SD) were calculated for each cycle. The mean represents the central tendency of participant responses, while the standard deviation reflects the dispersion of responses around that mean. However, unlike independent-samples t-tests, variability between groups is not treated as independent; instead, the analysis focuses on within-participant differences. Second, the difference score for each participant was computed as:

$$d_i = X_{i,C1} - X_{i,C2}$$

Equation 3.6: Calculating Difference

The mean of these difference scores ($\bar{d}$) represents the average change between cycles, while the standard deviation of the differences (SD_d) captures variability in individual-level change. Third, the standard error of the mean difference was calculated as:

$$SE_{\bar{d}} = \frac{SD_d}{\sqrt{n}}$$

Equation 3.7: Standard Error of Mean Difference

This statistic quantifies the expected sampling variability of the mean difference across participants. The t-statistic was then calculated using:

$$t = \frac{\bar{d}}{SE_{\bar{d}}}$$

Equation 3.8: T-Statistic

The degrees of freedom for the paired-samples t-test were calculated as:

$$df = n - 1$$

Equation 3.9: Degrees of Freedom for Paired Samples

With 20 participants, this resulted in *df* = 19. The calculated t-value was compared against the critical t-value at the chosen significance level ($\alpha = .05$). If the resulting p-value was less than .05, the difference was considered statistically significant.

3.4.4.3.2 Qualitative Methods

Open-ended questions (Table 3.10) used in the User Experience Insights (UEI) survey aim to capture participants' feedback on the baseline and modified systems. Feedback from Cycle 1 informs modifications to be implemented for Cycle 2. Feedback from Cycle 2 provides insights into the impact of these changes and offered guidance on potential further refinements, should additional cycles be conducted of the framework.

Table 3.10: Open-ended Questions

UEI1	What are your views on the * IDE?
UEI2	Is there any Areas that could still be improved? - Menu - Explorer - Editor - Terminal/Debugger - None
UEI3**	What further changes or improvements would you want to see implemented?

** - placeholder for baseline/modified, ** only valid if "None" was not selected in Q2*

3.4.4.3.2.1 Sample Quotations as Thematic Evidence

Qualitative responses were analysed using structured content analysis, with emphasis placed on preserving participant voice through the integration of sample direct quotations.

The thematic analysis conducted within this study should be interpreted as exploratory and descriptive rather than a formal standalone thematic-analysis procedure. Themes were derived through researcher-led review of participant responses to identify recurrent patterns, topics, and areas of concern relevant to the study objectives. Formal inter-coder reliability assessment and independent second-coder verification were not undertaken.

The inclusion of direct quotations serves a methodological rather than decorative function. Quotations provide evidentiary grounding for interpretive claims, allowing readers to trace analytical conclusions directly to raw data. This strengthens credibility, transparency, and confirmability within the qualitative strand. By retaining participants' original wording, the analysis preserves contextual nuance and semantic precision, particularly where responses include domain-specific terminology. Direct quotations therefore operate as analytic anchors that support thematic validity and interpretive rigour.

3.4.4.3.2.2 Word Cloud Analysis for Lexical Salience

Word cloud visualisation was employed as a descriptive lexical exploration and visualisation technique to examine patterns of word frequency and prominence within the qualitative dataset. Textual responses were cleaned, normalised, and processed to remove stop words and non-informative terms prior to generating frequency-based visual representations.

Word clouds were not used as a primary qualitative analysis method and were not interpreted in isolation. Rather, they served as visual aids for summarising vocabulary prominence and supporting the interpretation of themes identified through manual review of participant responses. Although this method is descriptive rather than inferential, it supports thematic interpretation by revealing patterns that may warrant deeper qualitative examination. In this study, lexical salience analysis functioned as a complementary layer to manual coding, enhancing transparency in how key terms and conceptual emphases were identified.

3.4.4.3.2.3 Word Cloud Analysis for Lexical Divergence

Lexical divergence analysis was conducted to examine variation in vocabulary distribution across textual corpora. This technique quantifies differences in word usage patterns between bodies of text, enabling structured comparison of semantic emphasis. By analysing relative frequency distributions and distinctive terminology, lexical divergence provides a measurable indicator of how language varies conditions (e.g. various cohorts).

In qualitative research, lexical divergence supports the identification of shifts in discourse patterns, including differences in affective tone, technical emphasis, or conceptual framing. Rather than replacing interpretive analysis, it augments it by introducing a comparative lexical metric that captures semantic variation at scale.

As with lexical salience analysis, lexical divergence was used as a descriptive exploratory technique intended to support qualitative interpretation rather than replace it. The resulting visualisations provide indications of differences in discourse patterns but should not be interpreted as evidence of thematic significance without consideration of the underlying participant responses.

3.4.4.3.2.3.1 Deriving Word Cloud

The word clouds presented were generated using a Python-based lexical processing pipeline (Figure. 3.3). The qualitative responses were exported as a plain text file and programmatically processed to remove punctuation, normalise terminology, and exclude stop words and non-informative terms. The cleaned corpus was then passed to the WordCloud library to compute term frequency and generate the visual representation.

```python
from pathlib import Path
import matplotlib.pyplot as plt
from wordcloud import STOPWORDS, WordCloud
import re
from collections import Counter

TXT_FILE = Path.cwd() / "filename.txt"

# Read text
text = open(TXT_FILE, mode="r", encoding="utf-8").read()

# 1 Normalisation

# Lowercase
text = text.lower()

# Standardise techno terms
text = text.replace("techno overload", "techno_overload")
text = text.replace("techno-overload", "techno_overload")
text = text.replace("techno complexity", "techno_complexity")
text = text.replace("techno-complexity", "techno_complexity")
text = text.replace("techno insecurity", "techno_insecurity")
text = text.replace("techno-insecurity", "techno_insecurity")
text = text.replace("techno invasion", "techno_invasion")
text = text.replace("techno-invasion", "techno_invasion")
text = text.replace("techno uncertainty", "techno_uncertainty")
text = text.replace("techno-uncertainty", "techno_uncertainty")

# Remove punctuation / numbers
text = re.sub(r"[^a-zA-Z_\s]", " ", text)

# 2 Stopword Expansion
custom_stopwords = {
    "ide", "modified", "version", "use", "using", "used",
    "also", "much", "make", "makes", "made",
    "task", "overall", "one", "way",
    "really", "quite", "lot", "well",
    "get", "got", "going",}

stopwords = STOPWORDS.union(custom_stopwords)

# 3 Token Filtering
tokens = text.split()
filtered_tokens = [word for word in tokens if word not in stopwords and len(word) > 2]
cleaned_text = " ".join(filtered_tokens)

# Inspect top frequencies before generating cloud
freq = Counter(filtered_tokens)
print(freq.most_common(20))

# 4. Generate Word Cloud
wc = WordCloud(
    background_color="white",
    height=400,
    width=1000)
wc.generate(cleaned_text)

# Store to file
wc.to_file("output.png")

# Display
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```
Figure 3.3: Source Code for Generating Word Clouds

3.4.4.3.3 Evaluation Metrics

In this study, measuring performance, productivity, and satisfaction is essential for addressing technostress in IDE usability. Prior research identifies technostress as a barrier to efficiency, engagement, and well-being, while the taxonomy analysis highlights a gap in understanding its impact on these factors in educational settings [276]. To bridge this gap, this study evaluates user experience through a combination of objective measures and subjective perceptions. Objective measures rely on quantifiable data, such as task completion time and test case success, while subjective measures capture participants' perceptions through self-reported surveys.

3.4.4.3.3.1 Task Outcomes

3.4.4.3.3.1.1 Performance

Performance refers to the effectiveness and efficiency with which users complete coding tasks within the IDE. Performance is assessed using four key instruments: code execution quality - in the form of Task Requirements (TRQ) and Task Test Cases (TTC), speed – in the form of Task Completion Time (TCT), PU, and SMI. Code execution quality is determined by evaluating whether the implemented solution meets the given requirements and test cases. Speed is measured by the time taken to complete the coding task, providing insights into efficiency. TAM > PU2 captured users' subjective evaluation of how beneficial they found the IDE in facilitating their coding tasks. Finally, SMI > O1 reflects users' evaluation of performance based on the modified IDE.

3.4.4.3.3.1.2 Productivity

Productivity refers to the extent to which users can efficiently complete their tasks while maintaining high-quality work. Productivity is assessed using three key instruments: Task Completion (TC), PU, and SMI. Task completion is an objective measure indicating whether participants successfully finished the assigned coding task within the given timeframe. TAM > PU3 serves as a subjective metric, capturing users' perspectives on how effectively the IDE supports their workflow and enhanced productivity. Finally, SMI > O2 reflects users' evaluation of productivity based on the modified IDE.

3.4.4.3.3.2 Satisfaction

Satisfaction refers to users' overall experience with the IDE, including how enjoyable and intuitive they find the system. Satisfaction is measured using IDE preference, SUS, PEOU, and SMI. IDE preference assessed whether users favoured the baseline or modified version of the IDE, providing insights into usability improvements. SUS measures users' overall perception of the system's usability. PEOU captured users' subjective experience regarding the simplicity and intuitiveness of the IDE's interface and functionalities. Finally, SMI > O3 reflected users' evaluation of satisfaction based on the modified IDE.

3.4.4.3.3.3 Technostress

Technostress refers to the negative psychological effects of using technology. Technostress was assessed using SMI survey items TS1–TS5, which captured participants' perceptions of techno-overload, techno-complexity, techno-invasion, techno-insecurity, and techno-uncertainty. These measures provided insights into how the modified IDE influenced participants' cognitive load and stress levels, highlighting whether system changes effectively reduced technostress triggers.

3.4.4.4 Modelling

At this stage, qualitative data is processed through fine-tuning Large Language Models (LLMs) to generate targeted Recommended Actions (RAs) for participant clusters, which are determined based on the area of the IDE where the modification was made. RAs represent specific, actionable system-level improvements derived from user feedback, such as adjustments to the IDE. LLMs can be used to generate concrete RAs elicited from the qualitative feedback and to map them to the user experience metrics they are intended to support. In Cycle 1, RAs are to be manually generated based on participant feedback to create an initial dataset, as the model required RAs for training, which will not be available at that initial stage. Since Cycle 1 feedback only provided insights on what needed to be changed, these manually generated RAs served as an initial input for training the model. From Cycle 2 onwards, participant feedback can be used in a continuous feedback loop to further refine and fine-tune these initial RAs, allowing the model to generate more targeted and effective recommendations in subsequent iterations.

3.4.4.5 Required Refinement

If the evaluation results indicate that no statistically or practically meaningful improvements are required, the iterative process concludes at this stage. The decision to terminate refinement is not based solely on subjective interpretation of feedback, but on predefined analytical criteria. Specifically, further modification is considered necessary only when (1) statistically significant differences are observed across cycles (e.g., $p < .05$ in performance, productivity, satisfaction, or technostress measures), (2) effect sizes indicate practically meaningful change, or (3) qualitative analysis reveals recurring usability or stress-related themes that remain unresolved.

Conversely, if quantitative measures suggest stability or plateauing improvement across cycles, confidence intervals show no meaningful variation, and qualitative feedback does not identify persistent technostress-related issues, the system is considered to have reached functional adequacy within the scope of the framework. Under these conditions, the iterative cycle is formally concluded. This approach ensures that refinement decisions are evidence-driven rather than subjective, aligning the framework with established experimental and HCI evaluation principles.

Within this framework, LLM-generated recommendations should be treated as participant refinement inputs rather than final system decisions. All outputs produced by the model should be subjected to a structured validation process in which recommendations are screened for alignment with participant feedback, technical feasibility within the IDE environment, and consistency with defined IDE functional categories. Outputs that are overly generic, redundant, or technically infeasible should be rejected or revised prior to inclusion in subsequent refinement cycles. In cases where LLM outputs conflict with participant feedback, participant-derived requirements should take precedence, and recommendations should be adjusted accordingly to maintain fidelity to the empirical evidence.

3.4.4.6 Modified IDE

If areas for enhancement are identified, modifications are introduced to refine the IDE experience. Based on the first cycle findings, specific modifications are implemented to enhance usability by addressing issues highlighted in participant feedback. These adjustments aim to improve the overall user experience in the second cycle. The second cycle follows the same data collection process as cycle one, but now with the modified IDE. Participants should complete alternative tasks under the revised conditions, allowing for a comparative analysis to assess the effectiveness of the modifications (hence addressing the research questions above). Following both cycles, a comprehensive analysis of qualitative and quantitative data (using the metrics outlined in earlier sections) identifies statistically significant changes in metric levels. These insights guide further refinements and adjustments, ensuring that modifications align with user needs. By identifying common patterns and challenges within specific groups, this step enables customised improvements that address diverse user needs through a data-driven approach. While this process is structured around cycles, further iterations may be necessary based on the findings. Additional refinement cycles revisit the study's objectives, introducing new tasks or expanding the participant pool to gather broader insights. By maintaining an iterative approach, the IDE continues to evolve based on comprehensive feedback and user-driven refinements. If resources allow, the process can extend indefinitely, with each cycle offering progressively refined improvements tailored to the evolving needs of users.

3.5 Framework Usage Illustrative Experiment

This section illustrates the practical application of the framework introduced in Section 3, demonstrating how it is operationalised across all three stages. This section also serves as the set up for evaluating the framework with results presented in Section 3.6.

3.5.1 Stage 1

This stage outlines the technical, procedural, and ethical foundations of the study. It begins by introducing the IDE selected for the experiment, followed by a summary of the ethical procedures adhered to throughout the study. Finally, the programming task assigned to participants is described, detailing how it was designed to evaluate their interaction with the IDE.

3.5.1.1 IDE Used

Visual Studio Code (VSCode) (v1.84.2) [277] which was used throughout participants undergraduate studies was also utilised for this experiment, ensuring consistent evaluation and reducing variability. In recent years, and continuing to the present, Stack Overflow's Developer Surveys (from 2021-2025) [278] consistently show that VSCode remains the most widely preferred IDE among developers. Its popularity is particularly strong among those learning to code, where adoption is even higher than among professional developers. The alignment between participant groups and real-world IDE usage strengthens the ecological validity of the study and ensures the findings remain applicable to both educational and professional development contexts. Aside from popularity several other factors such as accessibility, ease of use, and functionality [279].

Standard baseline VSCode comes with a simple and intuitive layout. The User Interface (UI) is split into several key areas [280].

(1) Menu: Provides access to essential commands and settings, including file management, editing tools, and configuration options
(2) Explorer: Displays the file structure of the project, which enables navigation of directories, opening files, and managing resources efficiently
(3) Editor: The main area for writing and editing code, featuring syntax highlighting, intelligent code completion, and support for various programming languages.
(4) Terminal/Debugger: An integrated terminal that includes output, error and warning messages, and debug information, enabling seamless execution and testing without leaving the IDE.

3.5.1.2 Ethical Considerations

Ethical approval for this project was granted by the Birmingham City University (BCU) Computing, Engineering and the Built Environment Faculty Academic Ethics Committee (CEBE FAEC) prior to any data collection taking place. The study was approved under reference number *Hassan /#9883 /sub1 /Fu /2021 /Oct on 08/Oct/2021*.

Participants within the School of Computing (SoC) were sent invitations, with participation being completely voluntary. Those who opted to partake in the study filled in a consent form. Prior to filling out the consent form, participants were provided with an Information Sheet outlining what the study was about, what it involved, and how they would be compensated for their participation and time. Additional procedures relating to the collection, anonymisation, storage, and protection of biometric data are described in Chapter 4.

3.5.1.2.1 Research Team Risks

The research team is comprised of myself, the lead researcher, under the supervision of the Director of Studies, Dr Sara Hassan, with Dr Andrew Wilson and Dr Shadi Basurra serving as the second and third supervisors respectively.

Operational Risks - The operational risks associated with conducting this research are minimal but acknowledged. As data collection involves programming-based experiments and evaluations in computing environments, risks such as technical faults, data loss, or participant scheduling challenges may arise. To mitigate these, all experimental data are securely backed up on encrypted institutional drives, and contingency plans are in place to replicate sessions in the event of technical failure. Clear communication protocols between the researcher and supervisors ensure timely coordination and minimise disruption to the research timeline.

Data Governance Risks - As the study involves human participants and biometric data, the research team bears ethical responsibility for maintaining confidentiality, informed consent, and data integrity. All members adhere to BCUs Ethics and GDPR guidelines. Regular data audits, anonymisation procedures, and restricted access to raw datasets ensure compliance with institutional and legal standards.

3.5.1.2.2 Participant Risks

While the study poses minimal risk, several potential factors have been considered to ensure participant wellbeing and data protection throughout.

*Cognitive and Emotional Risk*s - As the research focuses on technostress and mental wellbeing, there is a possibility that participants may experience mild cognitive fatigue, frustration, or stress during programming tasks, especially when encountering coding errors, time constraints, or complex IDE configurations. To mitigate these effects, all participants are briefed on the voluntary nature of their participation and are free to pause or withdraw at any point without penalty.

Technical and Environmental Risks - Minimal technical risks may arise from the use of computing equipment or IDEs during experimental sessions, such as temporary software crashes or system slowdowns. To address this, all sessions are conducted in well-supervised laboratory environments equipped with reliable hardware, and technical support is available throughout. Should unexpected issues occur, affected participants can repeat or reschedule sessions if deemed possible.

3.5.1.3 Coding Activity

The coding task for each cluster (refer to Appendix, Part B) was provided with a detailed description, outlining the objectives and context to ensure that participants clearly understood what they were expected to achieve. This was followed by a requirements section that specified the criteria and constraints of the solution, including all relevant technical specifications and guidelines. Additionally, a sample Input/Output (I/O) was included, giving examples of input data and the expected output to guide participants in achieving the functionality of their code effectively. This activity was validated against programming tasks which target participants complete within the undergraduate degrees to cater for the knowledge and skill levels. Before beginning the experiment, the researcher explained the activity to each participant. They were permitted to revisit the task document and browse the web to address any questions or obstacles encountered during the implementation phase.

3.5.1.3.1 Difficulty

The difficulty of the programming tasks was deliberately predefined and held constant across participants and experimental cycles. Tasks were designed to be representative of undergraduate-level programming activities, aligned with participants' prior training and coursework, and sufficiently complex to engage key IDE areas without introducing excessive difficulty that could confound technostress measurements. Both Task A and Task B were designed to require comparable programming knowledge, problem-solving effort, implementation complexity, and expected completion time. Furthermore, both tasks included a comparable number of functional requirements and accompanying test cases to ensure a similar level of development and testing activity. This design choice ensured that observed emotional and behavioural responses could be more confidently attributed to IDE design characteristics rather than variability in task complexity. However, no formal task-validation procedure or independent assessment of task equivalence was conducted prior to deployment. Consequently, although care was taken to match the tasks across key characteristics, it cannot be guaranteed that participants perceived them as identical in difficulty. As such, differences in task characteristics may have contributed to some of the observed differences between cycles and should be considered alongside potential order effects when interpreting the results.

3.5.1.3.2 Timeframe

The entire activity spanned a total of 18 minutes, divided into three distinct phases which align with the Software Development Life Cycle (SDLC) [281] (a) Analysis phase which lasted 3 minutes, this initial phase allowed participants to review the task requirements and plan their approach (b) Implementation and (c) Testing phases in which the remaining 15 minutes were allocated for writing code and testing solutions to ensure they met the specified requirements and produced the correct outputs.

The decision to allocate 15 minutes for the coding experiment was made to balance ecological validity, cognitive load, and participant engagement. Firstly, these durations align within the typical duration taken in labs when these students study Python. Secondly, data suggests that the average duration for programming tasks can hover around the 20 minutes range [282], though this can be heavily dependent on the task's complexity and the developer's experience. From a biometric perspective, a concise experimental window improves signal quality and interpretability. Prolonged exposure can often lead to emotional drift, whereas shorter sessions allow emotional and cognitive states to remain consistent throughout the task. Research by [283] found that after only 20 minutes of continuous or interrupted performance, participants experienced significantly higher levels of stress, frustration, workload, and perceived pressure. Setting the task duration just below this threshold ensured that participants remained engaged without inducing artificial stress or fatigue. This design choice helped preserve the ecological validity of the study, capturing authentic user interactions while maintaining data reliability across both behavioural and biometric measures.

3.5.1.3.3 Assistance

During the experiment, the primary researcher was present to explain the task, offering assistance as needed, and clarifying to the participants' their right to withdraw under the conditions detailed in the participant "Information sheet" detailing the ethical considerations accounted for within the experiment.

3.5.2 Stage 2

This stage provides a description of the participant demographics. The participant demographics subsection presents a detailed breakdown of the study sample, including attributes such as gender, education level, coding experience, and ethnicity. Together, these elements establish the foundational context for the study's methodology and ensure transparency and rigor in how the research was conducted.

3.5.2.1 Participant Demographic

This experiment was conducted using data acquired at BCU. The experiment included SoC students as research participants, ensuring a baseline of technical expertise relevant to the study. This aligns with HCI recommendations to study users who naturally operate within the interaction context being evaluated, thereby enhancing ecological validity while reducing confounding variability [284]. Twenty undergraduate participants were recruited (presented in Table 3.11), although the sample size is modest, this stage of the study is positioned as an exploratory investigation to support potential effectiveness, with threats to validity and generalisability discussed in Chapter 6 (section 6.2).

The gender ratio (4 females and 16 males) mirrors the current disparity between women and men in the tech industry. Recent trends indicate an upward trajectory for women's representation, with figures rising from 19% in 2019 to 26% in 2022 and reaching 29% in 2023 [285], this ratio is further reflected in the SoC at BCU, with 18% of students being females in the 24/25 academic year. The group consists of 10 first-year, 8 second year, and 2 third-year students. Regarding programming experience, 6 have less than 1 year, 10 possess 1-2 years, and 4 have over 2 years. Additionally, there are 15 Home/EU and 5 International students, with 11 of Asian, 3 of Black, 4 of White, and 2 of Other ethnicity. The distribution between the various ethnic groups matches the percentage of ethnicities and students there are in the SoC.

Table 3.11: Participant Demographic for Experiment

P	*Gender*	*Education*	*Experience*	*Ethnicity*
1	Male	Second Year	1-2 Years	Black
2	Male	Second Year	Over 2 Years	White
3	Female	First year	1-2 Years	Asian
4	Female	Second Year	1-2 Years	Asian
5	Male	First year	1-2 Years	Asian
6	Male	Second Year	Over 2 Years	Other
7	Male	Second Year	1-2 Years	White
8	Male	First year	Under 1 Year	Black
9	Male	First year	Under 1 Year	Asian
10	Female	First year	Under 1 Year	Asian
11	Male	Second Year	1-2 Years	Asian
12	Male	Second Year	1-2 Years	Asian
13	Male	Second Year	1-2 Years	Black
14	Male	First year	Under 1 Year	White
15	Male	Third Year	Over 2 Years	Asian
16	Male	Third Year	Over 2 Years	White
17	Female	First year	Under 1 Year	Asian
18	Male	Third Year	Over 2 Years	Asian
19	Male	First year	1-2 Years	Asian
20	Male	First year	1-2 Years	Other

3.5.3 Stage 3

The clustering stage explains how participants were divided into groups for A/B testing, ensuring a balanced distribution across tasks and reducing potential bias during evaluation. This is followed by an overview of the analysis methods adopted in each experimental cycle. The stage then details the processes involved in translating participant feedback into tangible IDE improvements. The RAs describes how participant feedback was analysed to identify potential interventions. The need for implementation is then discussed, outlining how insights from the previous stage informed modification decisions. This is followed by an explanation of how the RAs were implemented to modify the IDE and better meet user needs. Finally, the availability of the implemented modifications across other IDEs is considered, highlighting the transferability of the framework beyond VS Code.

3.5.3.1 Clustering

A within-subjects design was selected to enable direct comparison between the baseline and modified IDE while controlling for participant-specific differences such as programming experience, problem-solving ability, and emotional variability. By using the same participants across both cycles, the study increased sensitivity to changes introduced by the IDE modifications while reducing the influence of inter-participant variability. To further minimise task-related effects, participants were split into two clusters of ten (50:50 distribution). As illustrated in Figure 3.4, in Cycle 1, Cluster 1 completed Task A while Cluster 2 completed Task B. In Cycle 2, assignments were reversed, with Cluster 1 completing Task B and Cluster 2 completing Task A. This task alternation helped reduce the influence of task-specific factors and practice effects when comparing responses across cycles. However, as all participants completed Cycle 1 before Cycle 2, potential order effects cannot be completely excluded, and the absence of full counterbalancing remains a limitation of the study.

Participant grouping into Menu, Explorer, Editor, Terminal, and General clusters was based on thematic coding of qualitative feedback obtained from the UEI open-ended responses (Table 3.10). In addition to this, UEI2 provided participants with predefined IDE areas (Menu, Explorer, Editor, Terminal/Debugger, or None), which directly informed initial clustering decisions when explicitly selected. Responses indicating issues or suggestions clearly aligned with a single IDE functional area were assigned to the corresponding cluster. Where feedback was generic, ambiguous, or encompassed multiple areas without clear dominance, it was assigned to the General cluster, which represents cross-cutting or non-specific usability and experience issues. The same clustering approach was applied consistently across subsequent cycles to ensure comparability of participant grouping. The assignment process was conducted through researcher-led qualitative interpretation without independent coding or inter-rater verification, reflecting the exploratory nature of the clustering approach.

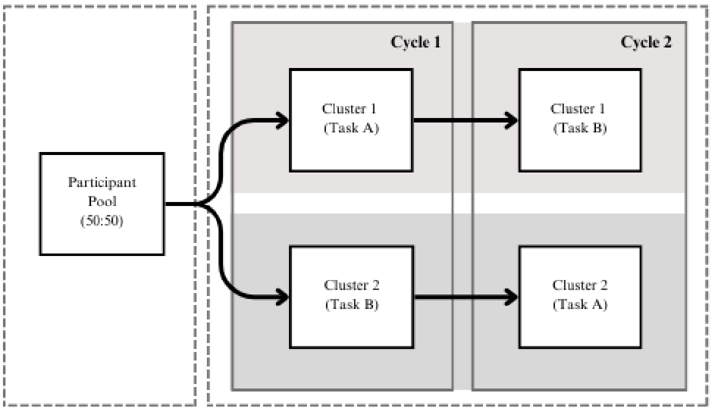


Figure 3.4: Analysis Approaches

3.5.3.2 Data Analysis

The adopted across the experimental cycles are summarised in Table 3.12, with detailed results presented in Section 3.4. Across all experimental cycles, including Cycle 1 and Cycle 2, SUS, TAM, and TO are consistently captured. These instruments collectively provide indirect but well-established indicators of technostress-related outcomes, specifically performance, productivity, and satisfaction, and enable longitudinal comparison across IDE configurations.

Cycle 1 represents interaction with a standard, unmodified IDE and serves as the baseline condition. Cycle 2 introduces targeted IDE modifications informed by empirical findings. The SMI survey is introduced from Cycle 2 onwards as an additional measurement layer, designed to capture explicit perceptions of technostress following design interventions. It is therefore not administered in Cycle 1, which precedes any system modification.

Importantly, RQ5 does not rely exclusively on SMI scores to assess technostress. Instead, technostress is operationalised through its outcome dimensions, performance, productivity, and satisfaction, which are measured consistently across all cycles using SUS, TAM, and TO metrics. The SMI complements these measures by providing direct insight into participants perceived technostress in modified IDE conditions, strengthening interpretation of whether observed changes in outcomes are aligned with reductions in technostress rather than general usability improvements alone.

Table 3.12: Approaches Adopted Across Cycles

Metric	***Category***	***Instrument***	***Cycle 1***	***Cycle 2***
Performance	Quantitative	Code Execution Quality (TO > TRQ, TTC)	x	x
		Speed (TO > TCT)	x	x
		Perceived Usefulness (TAM > PU2)	x	x
		System Modifications Impact (SMI > O1)		x
Productivity		Task Completion (TO > TC)	x	x
		Perceived Usefulness (TAM > PU3)	x	x
		System Modifications Impact (SMI > O2)		x
Satisfaction		IDE Preference		x
		System Usability Scale (SUS)	x	x
		Perceived Ease of Use (TAM > PEOU)	x	x
		System Modifications Impact (SMI > O3)		x
Technostress		System Modifications Impact (SMI > TS)		x
Experience	Qualitative	User Experience Impact (UEI)	x	x

3.5.3.3 Modelling

The qualitative feedback provided in Cycle 1 identified areas requiring changes or improvements for Cycle 2. Recommended Actions (RAs) were subsequently developed using a combination of manual generation and outputs produced by a Supervised Fine-Tuned (SFT) Large Language Model (LLM), ensuring that the final implementations remained closely aligned with participant needs. This approach allowed participant feedback to be systematically translated into actionable IDE modifications while maintaining researcher oversight throughout the process.

Based on their feedback, participants were assigned to clusters representing key functional areas of the IDE: Menu, Explorer, Editor, and Terminal. This clustering process ensured that modifications were targeted towards the areas most relevant to participant experiences and priorities. The SFT model served as a decision-support mechanism to assist in generating potential recommendations for each cluster, while final implementation decisions remained under researcher control.

3.5.3.3.1 Generating Recommended Actions

The tailored modifications implemented in this study were generated using GPT-3.5-turbo (Trained tokens 37130, Epochs 10, Batch size 1, LR multiplier 1). The fine-tuning process was carried out via the OpenAI Playground, using a custom dataset derived from qualitative user feedback. The dataset consisted of 20 feedback-to-recommendation pairs collected during Cycle 1 and was prepared in a .jsonl format (Figures 3.5 and 3.6) in accordance with OpenAI's fine-tuning guidelines [286]. Following the implementation of the resulting modifications, a further 20 feedback entries were collected during Cycle 2. These responses captured both additional recommendations for future improvements and participant evaluations of the changes introduced from Cycle 1, supporting the iterative refinement and validation of the recommendation-generation process.

Each training example contained participant feedback as the input and a corresponding recommended action as the desired output. Given the modest dataset size and the exploratory nature of the study, all available examples were used for fine-tuning, resulting in no separate validation or held-out test set being created.

OpenAI-provided training metrics, including loss values generated during the fine-tuning process, were monitored to ensure successful model convergence. As the objective of the model was to assist in generating participant recommendations rather than function as an autonomous decision-making system, evaluation focused primarily on the practical quality and relevance of generated outputs. Recommendations were assessed according to whether they accurately reflected participant feedback, aligned with the relevant IDE category (Menu, Explorer, Editor, or Terminal), and could be translated into feasible IDE modifications.

Human oversight was maintained throughout the process. All recommendations generated by the fine-tuned model were manually reviewed by the researcher before implementation. The SFT model functioned as a decision-support mechanism, with final implementation decisions remaining under researcher control. An example of a correct output is illustrated in Section 3.5.3.3.1.1 (Hypothetical Example), as no incorrect outputs were produced by the model in a retained or recorded form, since all generated recommendations were subject to manual review prior to implementation.

A limitation of this approach is that no formal held-out testing procedure was conducted. Therefore, the model's performance should be interpreted as a practical proof-of-concept for recommendation generation rather than a rigorously benchmarked machine-learning system.

```
{"messages": [
  {"role": "system", "content": "System-level instructions providing model context"},
  {"role": "user", "content": "Qualitative user feedback"},
  {"role": "assistant", "content": "Recommended Action (RA) generated in response"}]}
```

Figure 3.5: GPT .Json Dataset Structure

```
{"messages":[
{"role": "system", "content": "You are a helpful assistant that classifies user feedback on IDEs and suggests improvements for more Wellbeing Integrated Development Environments (TIDEs)."},{"role": "user", "content": "If it showed you a summary of the error as you hover over it"}, {"role": "assistant", "content": "🔧 Recommended Action: Install or configure an extension that enhances inline error tooltips to display concise summaries or explanations on hover\n📁 Category: Editor\n😊 Sentiment: Neutral\n🎯 Outcome: Improved Satisfaction (more clarity and ease during debugging)\n😮‍💨 Reduces Technostress: Techno-invasion (by minimizing the need to constantly switch focus or seek external explanations; improved syntax support and visual clarity help users stay in flow and reduce error-related frustration)"}]}
```

Figure 3.6: GPT .Json Dataset Example Sample

The training procedure was stable and successful (Figure 3.7) , as indicated by the continuous decline in loss values and the corresponding rise in accuracy throughout the training iterations. The loss curve indicates a steady reduction from approximately 3.2 to 1.0, while the accuracy curve exhibited a consistent improvement from around 0.45 to 0.9, indicating effective learning and convergence without overfitting. The fine-tuning results support that the model is capable of generalising within its domain and producing coherent, context-aware responses aligned with the TIDEs research objectives.

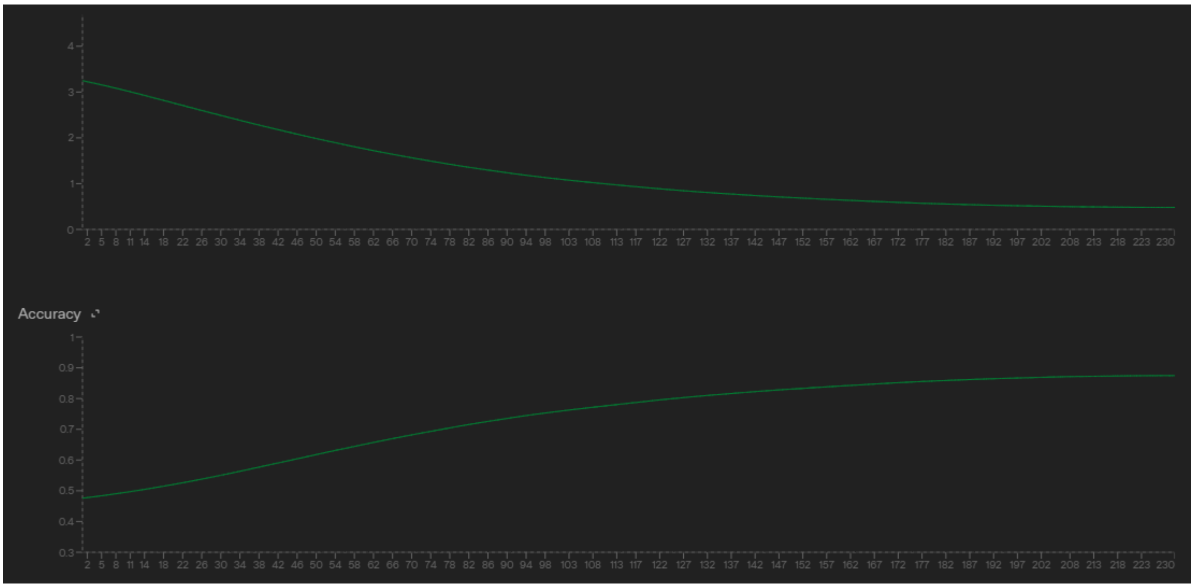


Figure 3.7: SFT Model Training Loss and Accuracy

The outputted SFT model, *ft:gpt-3.5-turbo-0125:wides:wides-model-v3-5-turbo:BOpojYKt*, can be accessed and utilised through the API for automated feedback analysis, classification, and recommendation generation. Furthermore, the model is designed to be iteratively refined as additional data is collected across future experimental cycles. This enables continuous learning and adaptation, allowing the model to evolve alongside the expanding TIDEs dataset to enhance its accuracy, contextual understanding, and capability in supporting technostress-aware IDE development.

3.5.3.3.1.1 Hypothetical Example

Initially, RAs would be manually curated by the researcher. Following Cycle 1, these would be validated through user responses and adjusted as needed. This iterative refinement should ensure the generated recommendations remained relevant, actionable, and aligned with user needs while promoting a reduction in technostress triggers, with the addition of the trigger which potentially could be increased, obtained through iterative feedback of the implemented modification.

To illustrate the model's usage, once the dataset has been validated and the model trained, it would be provided with a system message that establishes contextual boundaries and defines how responses should be structured. As shown in Figure 3.8, the model processes user feedback and outputs a structured response that includes: (1) a recommended action, specifying what modification should be implemented within the IDE; (2) a categorisation of the relevant IDE area to which the modification applies; (3) a sentiment classification, identifying the tone of the feedback as positive, negative, or neutral; (4) the intended outcome, indicating whether the recommendation primarily enhances performance, productivity, or satisfaction; and (5) the predominant technostress trigger likely to be reduced as a result of the recommended modification.

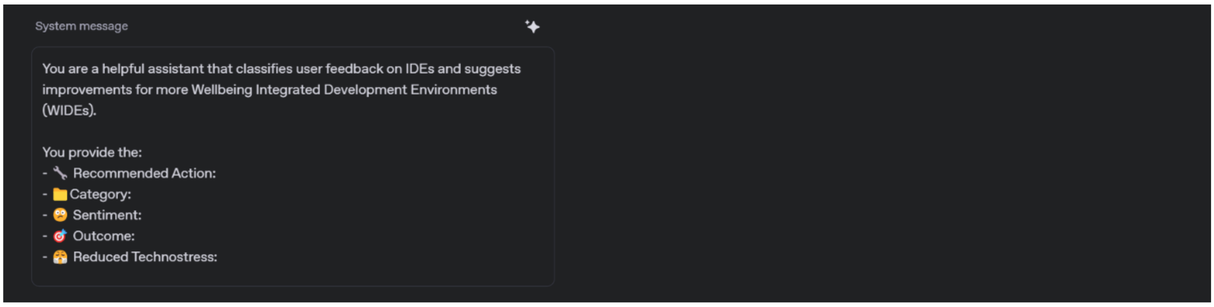


Figure 3.8:Providing System Message to SFT Model

Figure 3.9: Testing the SFT Model by Providing Input

The model then generates the appropriate output based on its training. The model (Figure 3.10) identifies the RA as implementing an extension to enhance code autocompletion. This recommendation would aligns with the Editor region of the IDE, and the tone of the feedback is classified as neutral. The expected outcome of implementing this RA is an improvement in performance, as enhanced autocompletion supports faster coding and reduces the likelihood of syntax or logical errors. Consequently, this modification will primarily help to mitigate techno-uncertainty by increasing user confidence in code accuracy through smarter, context-aware suggestions.

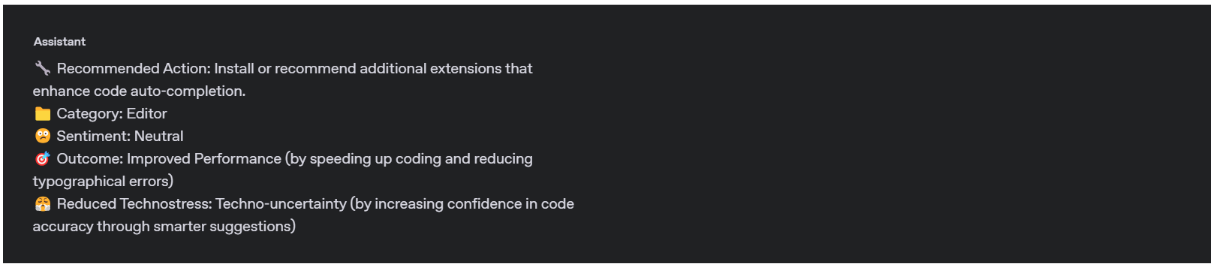


Figure 3.10: Output Provided by SFT Model for Cycle 2

In subsequent iterative cycles, as the training dataset expands and becomes more refined, the accuracy and contextual relevance of the RAs are also expected to improve. Qualitative feedback collected during Cycle 2 would indicate that the initial RA, introducing enhanced code autocompletion was largely effective and correctly identified. However, participants noted that while this modification successfully reduced techno-uncertainty by increasing confidence in code accuracy, it could also contribute to techno-overload if the autocompletion became overly intrusive or generated irrelevant suggestions (Figure 3.11).

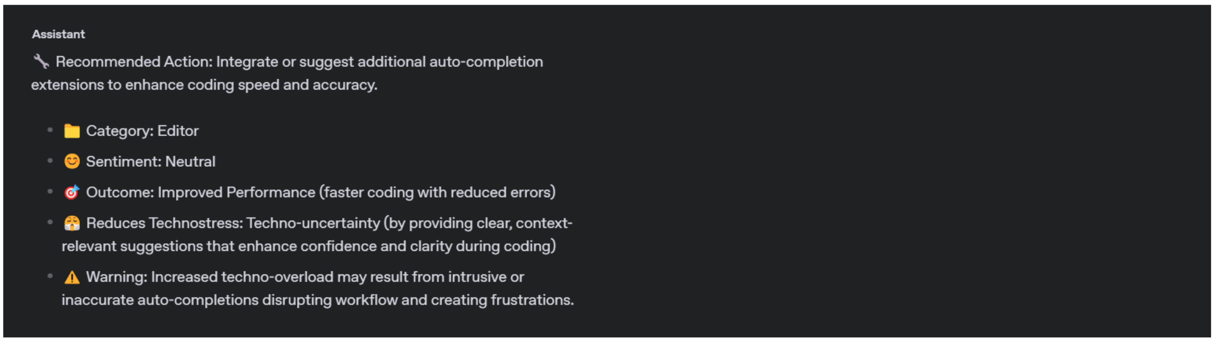


Figure 3.11: Refined Output Provided by SFT Model for Further Iterative Cycles

The iterative process will therefore not only enabled the refinement and adjustment of the initial recommendation where necessary but has also provided deeper insights into its potential drawbacks. These findings would emphasise the value of continual feedback .

Importantly, this iterative feedback would inform developers responsible for implementing such changes, guiding them on what aspects to prioritise and what potential side effects to consider. By identifying both the benefits and limitations of specific modifications, the process ensures that future IDE enhancements are not only technically effective but also aligned with user wellbeing, promoting a balance between support, autonomy, and cognitive comfort.

3.5.3.4 Required Refinement

As a result of the modelling step, it makes it possible to identify clusters of feedback associated with specific Areas of Interest (AOIs). The three key AOIs identified were the Menu, Editor, and Terminal, along with some general feedback. In several instances, participants provided multiple recommendations for the same area, resulting in overlapping suggestions across clusters. The use of a LLM helped to extract, consolidate, and cluster similar recommendations from different participants. This process ensured that common patterns were identified efficiently and that the resulting RAs accurately reflected collective user needs, which in turn guided the modifications implemented in the IDE.

3.5.3.5 Modified IDE

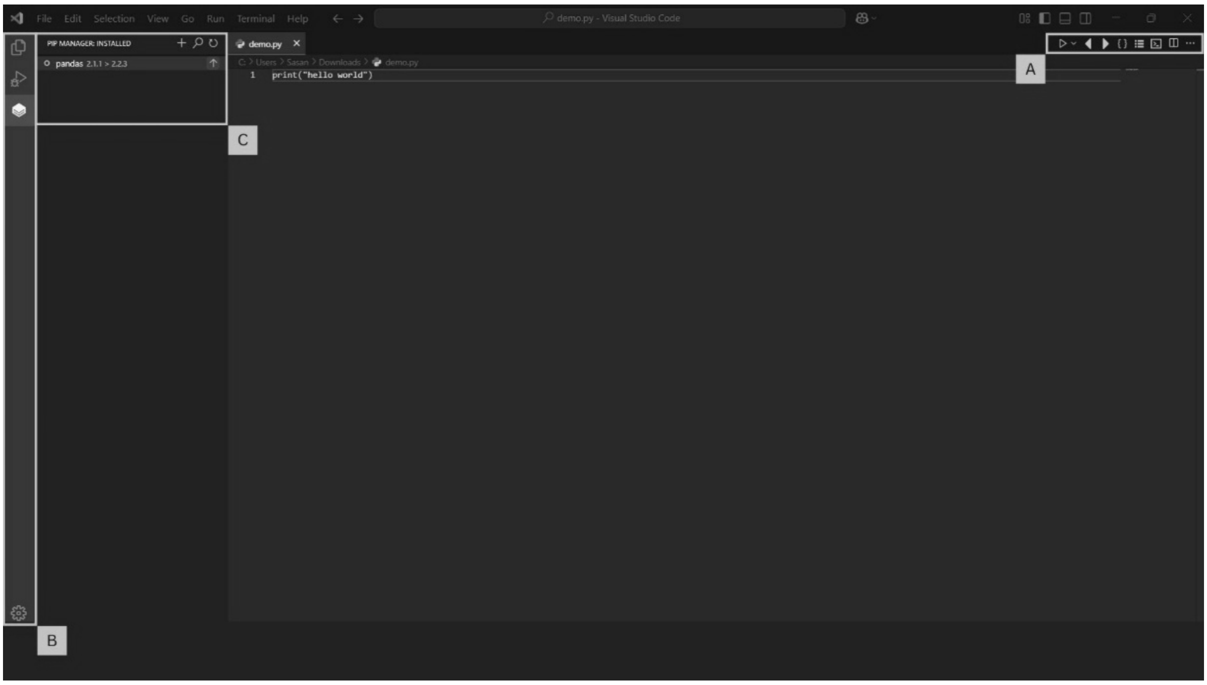
The modifications were selected because they addressed recurring challenges observed during Cycle 1. Collectively, these changes were chosen to operationalise the RAs derived from qualitative feedback (section 3.3.3.3) and to directly target areas identified.

3.5.3.5.1 Menu Cluster Modifications

For the **Menu**, key modifications (Table. 3.13) which were implemented (Figure. 3.12) to be assessed were:

Table 3.13: Menu Modification List

Mod	***Name***	***Description***	***Sample Response***
A	*Shortcut Menu*	Introduced to address participant feedback indicating difficulty in locating frequently used functions and frustration with repetitive navigation. Providing quick access to common actions such as *Save, Save All, Navigate Back/Forward, Beautify/Format Document, Show/Hide Terminal, Toggle Activity Bar, and Find & Replace* aimed to enhance efficiency and reduce cognitive effort. Allowing up to ten user-defined buttons further supported personalisation and autonomy.	P18 “The only thing that may need changes would be the help button, or maybe getting use to of how to run the code It wasn't straight forward to find.”
B	*Simplified Navigation*	Implemented in response to reports of visual clutter and disorientation when switching between views. This change aimed to streamline the interface layout, aiming to improve intuitiveness and helping users focus on task-relevant components.	P8 “cleaner UI e.g activity bar”
C	*Package Manager*	Added following comments highlighting the difficulty of managing dependencies through command-line operations. By allowing users to view, update, and install packages directly within the IDE interface, this feature aimed to reduce reliance on terminal commands, which several participants associated with increased frustration and task interruption.	P5 “see what packages are installed without needing to use commands.”



Key: A - Shortcut Menu, B - Simplified Navigation, C - Package Manager

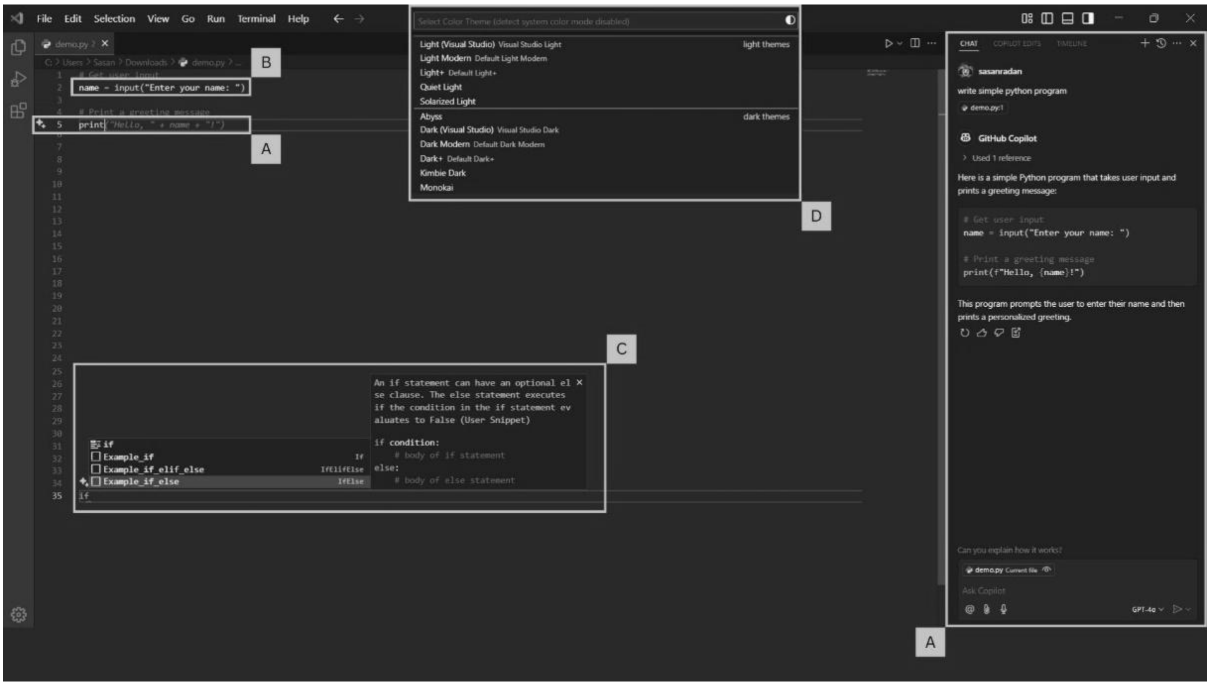
Figure 3.12: Menu Modifications Implemented

3.5.3.5.2 Editor Cluster Modifications

For the Editor key modifications (Table 3.14) implemented (Figure. 3.13) for assessment.

Table 3.14: Editor Modification List

Mod	*Name*	*Description*	*Sample Response*
A	*Auto-Completion*	Introduced in response to feedback noting repetitive typing and uncertainty about syntax correctness, the auto-completion feature provided intelligent, context-aware suggestions. This enhancement aimed to streamline the coding process, minimise typographical errors, and reduce time spent recalling syntax details.	P14 “I think the experience of VSCode can be certainly improved with 3rd party plugins such as GitHub Copilot”
B	*Syntax Highlighting*	Enhanced through bracket colourisation and clearer visual separation of code blocks, this modification addressed participants’ comments about difficulty in navigating nested code structures. Improved readability supports faster debugging and comprehension, particularly under time constraints.	P12 “Adding colours to brackets to easily separate the different inner brackets e.g. rainbow brackets extensions”
C	*Built-in Templates/Snippets*	Developed to respond to users expressing difficulty in recalling correct structures or starting points for implementation tasks. These templates included both generic snippets (e.g., functions, conditionals, and error-handling) and task-specific snippets tailored to the assigned exercises. The inclusion of explanatory comments alongside each snippet aided comprehension and learning.	P7 “Some quick-to-use templates to reduce the amount of time it takes to code a solution.”
D	*Themes library*	Expanded based on feedback related to visual fatigue and discomfort during extended sessions. Offering a broader range of light and dark themes allowed participants to select an interface that matched their visual preference and lighting conditions.	P16 “Theme wise, although I prefer dark mode, it would nice to see some more themed options or just a way to make the systems more engaging.”



Key: A - Auto-Completion, B - Syntax Highlighting, C - Built-in Templates/Snippets, D - Themes library

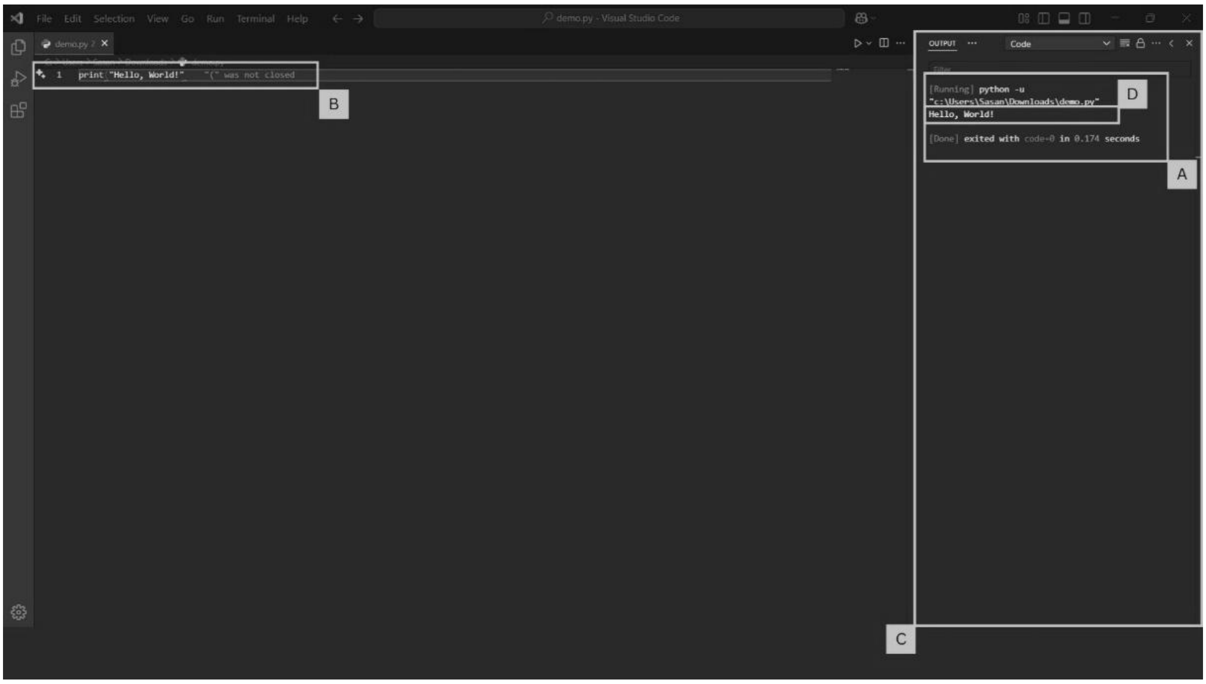
Figure 3.13: Editor Modifications Implemented

3.5.3.5.3 Terminal/Debugger Cluster Modifications

For the **Terminal/Debugger**, key modifications (Figure 3.15) which were implemented (Figure. 3.14) to be assessed were:

Table 3.15: Terminal/Debugger Modification List

Mod	*Name*	*Description*	*Sample Response*
A	*Runtime Indicator*	Introduced to provide clear visual feedback on code execution, displaying when the program started, when it ended, and the total runtime duration. Participants in Cycle 1 often mentioned uncertainty about whether their program was still running or had completed execution, which caused confusion and wasted time re-running scripts.	P2 "Faster runtime when "run" button is pressed or a way for me to know that the code is being compiled or interpreted"
B	*Error Handling*	Enhanced with real-time tooltips showing detailed error explanations upon hovering over issues. Feedback from participants indicated that errors were a primary source of frustration and cognitive interruption. The improved system aimed to reduce the need to switch contexts or search online for error meanings, fostering smoother debugging.	P2 "If errors are shown before execution to lessen error handling when it comes to debugging"
C	*Layout*	The terminal's position was adjusted from the default bottom panel to the right-hand side of the interface. This redesign emerged from feedback suggesting that the original layout constrained screen space and required frequent toggling between code and output. The right-hand placement was intended to optimise the workspace, enabling simultaneous viewing of code and execution results.	P10 "The layout of the software VSCode, the results of the terminal shown at the bottom it would be better if it was at the side"
D	*Terminal Output*	Refined to remove redundant or verbose text, resulting in a cleaner, more focused display. Participants noted that excessive output lines often obscured meaningful information, particularly during debugging.	P13 "seems more cluttered when you are not using a separate monitor"



Key: A - Runtime Indicator, B- Error Handling, C – Layout, D - Terminal Output

Figure 3.14: Terminal/Debugger Modifications Implemented

3.5.3.5.4 General Cluster Modifications

Feedback from Cycle 1 revealed that none of the participants identified the **Explorer** as an area requiring improvements. As a result, no modifications were made prior the Explorer in Cycle 2. Participants who provided more generic feedback, without specifying a particular area for improvement, were placed into a "General" cluster. These participants experienced a combination of modifications implemented across the three primary clusters- Menu, Editor, and Terminal/Debugger. This approach ensured that even those with broader or non-specific feedback were exposed to meaningful enhancements, allowing for a comprehensive evaluation of the modifications and their potential impact on usability and user satisfaction.

3.5.3.5.5 Modification Availability and IDE Limitations

A comparative analysis (Table 3.16) of modification availability across multiple IDEs revealed that VSCode offers the highest degree of flexibility, making it the most effective environment for implementing the intended enhancements. Moreover, several of the modifications developed for VSCode were found to be transferable to other IDEs to varying extents.

VSCodes extensive extension marketplace enables seamless integration of a wide range of modifications. In contrast, IDEs such as Spyder lack any formal plugin marketplace, preventing comparable modifications from being introduced. While Eclipse and IntelliJ provide plugin repositories, the availability and reliability of required extensions remain inconsistent, with some still in beta or requiring substantial manual configuration. Key modifications, such as a customisable shortcut bar, integrated package manager, code assist, syntax highlighting, built-in templates and snippets, bracket colourisation, and themes are readily accessible in VSCode either natively or through stable extensions. Conversely, IDEs like Vim and Notepad++ often require custom implementations, significantly complicating integration. VSCode also supports flexible modification of runtime indicators, error tooltips, and terminal layouts using native settings or marketplace plugins. While IntelliJ and Eclipse offer partial alternatives, they lack built-in mechanisms for streamlined terminal customisation, typically requiring multiple plugin dependencies or custom scripting.

Although some modifications can be replicated in other IDEs through plugins or manual configuration, their overall feasibility remains limited. Eclipse, for example, suffers from plugin stability issues frequently reported by users. Similarly, Vim and Notepad++ rely on external scripts or manual adjustments to achieve functionality available natively in VSCode. Atom was excluded due to its official discontinuation, further reducing the number of viable alternatives.

Table 3.16: Modification Availability Across Various IDEs

		Menu			Editor				Terminal			
	Marketplace	a	b	c	a	b	c	d	a	b	c	d
VSCode	Y[287]	**[288]	*	**[289]	**[290]	**[291]	***	*/**	**[292]	**[293]	*	**[292]
Eclipse	Y[294]	-	*	-	**[295]	~[296]	~[297]	**	-	~[298]	*	-
IntelliJ	Y[299]	-	*	-	**[300]	**[301]	-	*/**	-	-	*	-
Spyder	N	-	*	-	-	-	-	*	-	-	*	-
#Atom	Y	-	-	-	-	-	-	-	-	-	-	-
Vim	Y[302]	-	-	-	-	-	~[303]	~[304]	-	-	*	-
Notepad++	Y[305]	-	*	-	-	-	*	*	-	-	-	-

* = Can be directly modified within IDE, ** = requires installation from marketplace, *** = custom implementation
~ = alternative available, - = no alternative available, # = excluded, (a)-(d) = modifications referred to in earlier subsections

3.6 Framework Evaluation

In this evaluation section, the headings of performance, productivity, satisfaction, and technostress are not used directly as subsection titles, even though they constitute the primary evaluation metrics of the framework. Instead, the subsections are organised according to the measurement instruments presented in Table 3.12.

This structural decision was made to ensure that all elements derived from the same instrument are discussed collectively, allowing for methodological coherence and clearer interpretation of instrument-specific findings. Grouping results in this way preserves the internal validity of each measurement construct and avoids fragmentation across multiple metric-based sections.

However, in Section 3.7, the findings are synthesised and explicitly discussed in relation to the overarching evaluation metrics of performance, productivity, satisfaction, and technostress. That later section provides a consolidated interpretation of the results with respect to the framework's intended evaluation dimensions.

3.6.1 System Usability and Technology Acceptance

3.6.1.1 Overall Cohort Responses

Of the participants who took part in Cycle 1 of the study (n = 20), the overall SUS score for this cohort (M = 78), indicating good usability, showed moderate variability (SD = 12) according to standard SUS benchmarks. This suggests that participants generally found the baseline IDE to be above average in usability. The same cohort of participants was asked to partake in Cycle 2, allowing for a comparison to their previous experience and an evaluation of whether the modifications made to the IDE, based on their previous suggestions, improved their experience in relation to the standard baseline IDE in Cycle 1 (Figure. 3.15). In Cycle 2, the SUS score increased (M = 85), demonstrating an improvement in usability, and this was accompanied by a slight increase in variability (SD = 15). While the increase suggests that participants responded positively to the modifications introduced, these findings should be viewed as an encouraging indication of improved usability rather than definitive evidence of a technostress-focused design improvement. Additional iterative design and evaluation cycles would be required to establish stronger evidence regarding the long-term impact of the modifications on user experience and technostress reduction. While the SD slightly increased compared to Cycle 1, this variability is expected due to the diverse opinions arising from the range of modifications introduced. The SEM was calculated to assess the precision of the estimated cohort responses. For Cycle 1, the SEM was 2.80, while for Cycle 2, it was 3.38, indicating slightly greater variability in the estimate of the mean for Cycle 2. The non-overlapping SEM ranges between the two cycles (Cycle 1: 75.45–81.05; Cycle 2: 82.12–88.88) suggest that the difference in mean scores is potentially statistically significant, indicating a meaningful improvement in overall participant responses from Cycle 1 to Cycle 2.

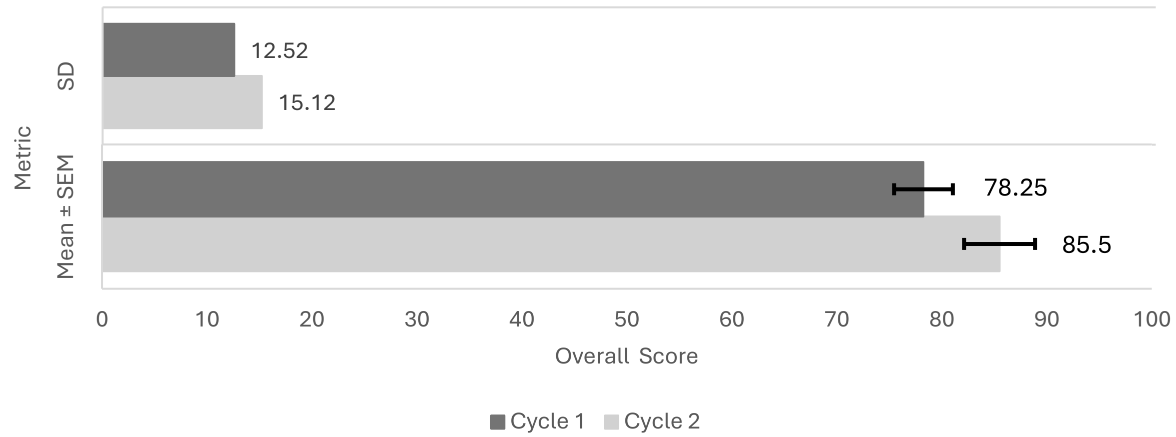
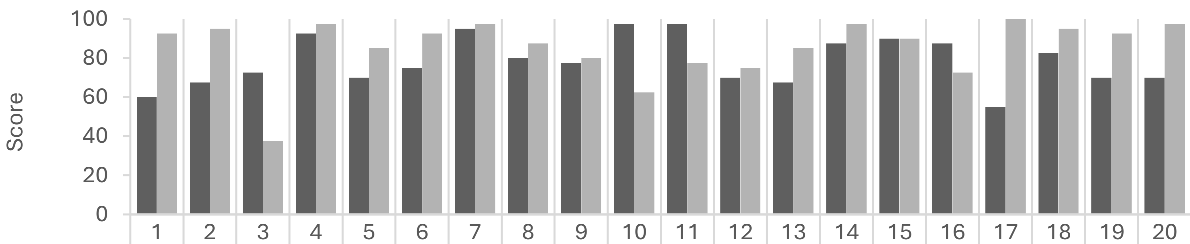


Figure 3.15: SUS Overall Cohort Score

Of the 20 participants who took part 16/20 (80%) scored the modified IDE of Cycle 2 higher than the baseline IDE of Cycle 1 (Figure. 3.16).



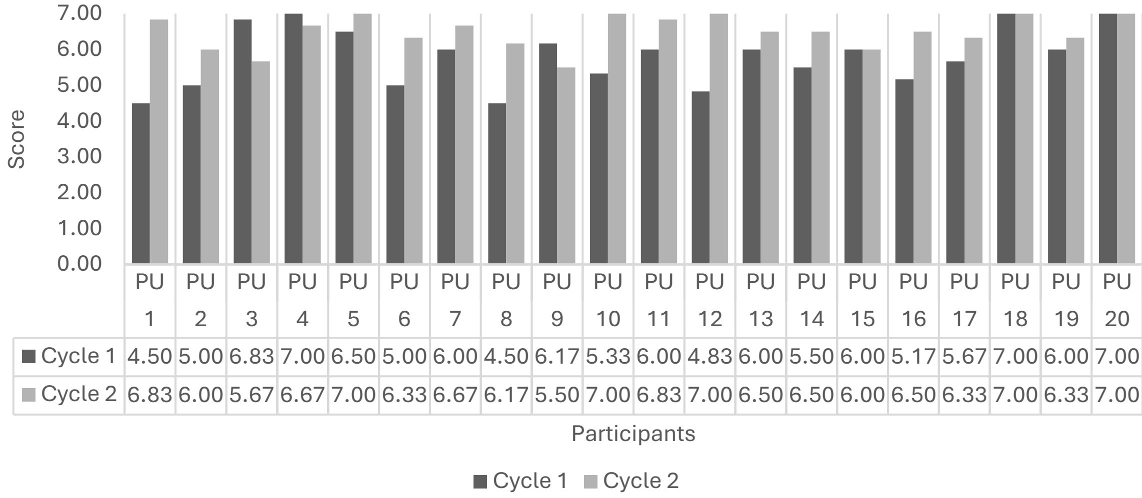
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Cycle 1	60	67.5	72.5	92.5	70	75	95	80	77.5	97.5	97.5	70	67.5	87.5	90	87.5	55	82.5	70	70
Cycle 2	92.5	95	37.5	97.5	85	92.5	97.5	87.5	80	62.5	77.5	75	85	97.5	90	72.5	100	95	92.5	97.5

Participants

Cycle 1 Cycle 2

Figure 3.16: SUS Overall Individual Score

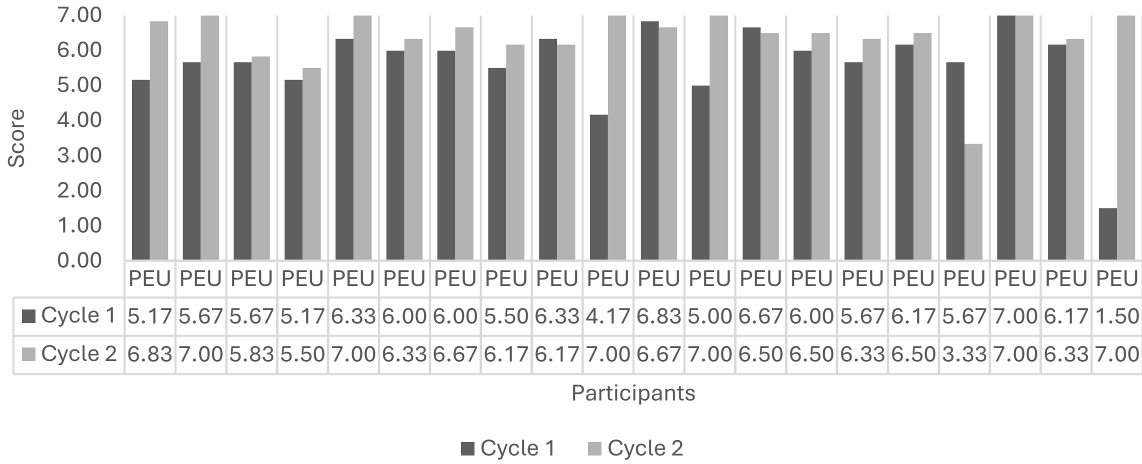
As for TAM, there were similar trends. Of the 20 participants who took part, 17/20 (85%) scored the modified IDE of Cycle 2 higher than the baseline IDE of Cycle 1 for PU (Figure. 3.17).



	PU 1	PU 2	PU 3	PU 4	PU 5	PU 6	PU 7	PU 8	PU 9	PU 10	PU 11	PU 12	PU 13	PU 14	PU 15	PU 16	PU 17	PU 18	PU 19	PU 20
Cycle 1	4.50	5.00	6.83	7.00	6.50	5.00	6.00	4.50	6.17	5.33	6.00	4.83	6.00	5.50	6.00	5.17	5.67	7.00	6.00	7.00
Cycle 2	6.83	6.00	5.67	6.67	7.00	6.33	6.67	6.17	5.50	7.00	6.83	7.00	6.50	6.50	6.00	6.50	6.33	7.00	6.33	7.00

Figure 3.17: TAM PU Overall Individual Score

As for PEU 16/20 (80%) scored the modified IDE of Cycle 2 higher than the baseline IDE of Cycle 1 (Figure. 3.18).



	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU	PEU
Cycle 1	5.17	5.67	5.67	5.17	6.33	6.00	6.00	5.50	6.33	4.17	6.83	5.00	6.67	6.00	5.67	6.17	5.67	7.00	6.17	1.50
Cycle 2	6.83	7.00	5.83	5.50	7.00	6.33	6.67	6.17	6.17	7.00	6.67	7.00	6.50	6.50	6.33	6.50	3.33	7.00	6.33	7.00

Figure 3.18: TAM PU Overall Individual Score

3.6.1.2 Breakdown by Statement

The initial scores from Cycle 1 across the ten statements/survey questions suggest that participants generally found the system effective and user-friendly. However, the relatively high standard deviation indicates variability in individual experiences. While many participants had a positive experience, others encountered difficulties, reflecting a range of interactions with the system. Scores from Cycle 2 (Figure. 3.19), indicate there were improvements across most statements, including S1, S3, S4, S5, S6, S7, S9, and S10. In Cycle 2, S2 and S8 either remained unchanged or experienced a slight decline compared to their baseline score from Cycle 1, indicating no noticeable improvement.

Examining the individual questions in detail reveals nuanced insights into these improvements. Participants' desire to use the system frequently, captured by **S1**, underscores its appeal. In Cycle 1, the mean score (M = 4.20), reflecting strong interest in frequent use, showed moderate variability (SD = 0.951). In Cycle 2, the mean increased (M = 4.45), demonstrating heightened enthusiasm, while the responses showed greater consistency (SD = 0.76). This improvement aligns with S1's upward trend, reflecting the system's enhanced usability and appeal in Cycle 2. The increase supports that modifications made the IDE more engaging, further encouraging regular use and solidifying participants' preference for the evolved system. **S2** examines perceptions of unnecessary complexity. In Cycle 1, the mean score (M = 1.50), indicating that participants found the baseline system straightforward, was paired with low variability (SD = 0.607). In Cycle 2, the mean increased slightly (M = 1.65), suggesting a marginal rise in perceived complexity, while variability also increased (SD = 0.93). This aligns with the lack of improvement seen in S2 and highlights the trade-off introduced by new features, which may have appeared more sophisticated to some participants. Nevertheless, the slight increase in complexity did not significantly detract from the overall user experience, as participants still rated the system as largely simple and accessible. **S3** explores the system's ease of use. In Cycle 1, the mean score (M = 4.15), which indicated positive perceptions of user-friendliness, was supported by relatively low variability (SD = 0.813). In Cycle 2, the mean improved further (M = 4.40), signalling greater ease of use, though variability increased slightly (SD = 0.88). This reflects the improvement observed in S3 and supports that the modifications enhanced the system's intuitiveness, allowing participants to interact with it more effortlessly. The increase in ease of use underscores the success of the enhancements in addressing user needs while maintaining simplicity. The need for technical support, assessed by **S4**, highlights participants' comfort levels with the system. In Cycle 1, the mean score (M = 1.65), suggesting most participants did not require external help, was accompanied by notable variability (SD = 1.040). In Cycle 2, the mean decreased (M = 1.50), indicating a slight improvement, though variability remained high (SD = 1.10). This aligns with S4's trend and indicates that the modifications improved participants' understanding of the system. Despite the improvement, the persistent variability suggests that additional guidance or onboarding features could further reduce the perceived need for support, particularly for less experienced users. **S5** evaluates how well the system's functionalities are integrated. In Cycle 1, the mean score (M = 4.00), reflecting participants' general satisfaction with functionality integration, was paired with low variability (SD = 0.725). In Cycle 2, the mean increased (M = 4.30), demonstrating improved perceptions of integration, while variability decreased (SD = 0.66). This aligns with S5's positive trend and highlights the success of the modifications in creating a more seamless and cohesive system. **S6** assesses the system's consistency. In Cycle 1, the mean score (M = 1.75), indicating perceptions of consistency, showed notable variability (SD = 1.020). In Cycle 2, the mean decreased slightly (M = 1.65), reflecting improved design consistency, though variability remained relatively unchanged (SD = 1.04). This aligns with S6's upward trend and suggests that the modifications addressed some inconsistencies. However, the remaining variability indicates that further refinements could enhance the system's reliability and predictability. **S7**, which measures how quickly participants could learn the system, showed significant improvement. In Cycle 1, the mean score (M = 3.80), indicating positive perceptions of learnability, was paired with moderate variability (SD = 0.951). In Cycle 2, the mean increased substantially (M = 4.45), indicating easier learning, with reduced variability (SD = 0.69). This aligns with S7's strong improvement and highlights the effectiveness of the modifications in simplifying the learning process. Participants reported that the evolved system was easier to understand and navigate, contributing to enhanced usability. **S8** examines whether participants found the system cumbersome. In Cycle 1, the mean score (M = 1.60), reflecting minimal perceptions of cumbersomeness, showed low variability (SD = 0.598). In Cycle 2, the mean remained the same (M = 1.60), though variability increased (SD = 1.14). This stagnation aligns with the lack of improvement in S8, suggesting that while most participants continued to find the system manageable, some perceived the added features as cumbersome. These results highlight an area for potential refinement to streamline the system further. **S9** evaluates participants' confidence in using the system. In Cycle 1, the mean score (M = 4.15), reflecting high confidence levels, showed moderate variability (SD = 0.875). In Cycle 2, the mean increased significantly (M = 4.65), indicating greater confidence, with a reduction in variability (SD = 0.75). This corresponds to S9's notable improvement and suggests that participants felt more assured in their ability to use the modified system effectively. The enhancements clearly empowered users and contributed to their overall satisfaction. **S10** examines perceptions of the system's learning curve. In Cycle 1, the mean score (M = 2.50), indicating manageable but somewhat challenging learnability, showed high variability (SD = 1.318). In Cycle 2, the mean decreased (M =

2.10), reflecting a reduction in perceived difficulty, with lower variability (SD = 0.93). This improvement aligns with S10's significant upward trend and highlights the success of the modifications in creating a more accessible and intuitive system for users with varied levels of experience.

The SUS statements that indicate the greatest notable gains were S7, S9, and S10. These results suggest that, while participants might have perceived the evolved system as somewhat more complex compared to the baseline (S2), yet they still viewed the modified IDE as easier to learn and navigate for most users (S7). This perception of enhanced usability further contributed to a higher level of confidence in using the system (S9), as from participants' views the learning curve was far less great before they could get going with the modified system (S10). In other words, although the changes introduced some additional complexity, participants felt more capable and assured when interacting with the modified IDE, which positively influenced their overall experience and confidence, positioning the modified IDE as a significant step forward.

Across most statements, SEM values in Cycle 2 were comparable to or lower than those in Cycle 1, indicating stable or improved precision in the estimated means following the IDE modifications. For several items (S1, S3, S7, and S9), the SEM ranges between Cycle 1 and Cycle 2 did not overlap, implying that the observed increases in mean scores are potentially statistically significant. Overall, the SEM analysis supports the interpretation that changes in SUS responses from Cycle 1 to Cycle 2 reflect meaningful and reliable improvements rather than sampling variability.

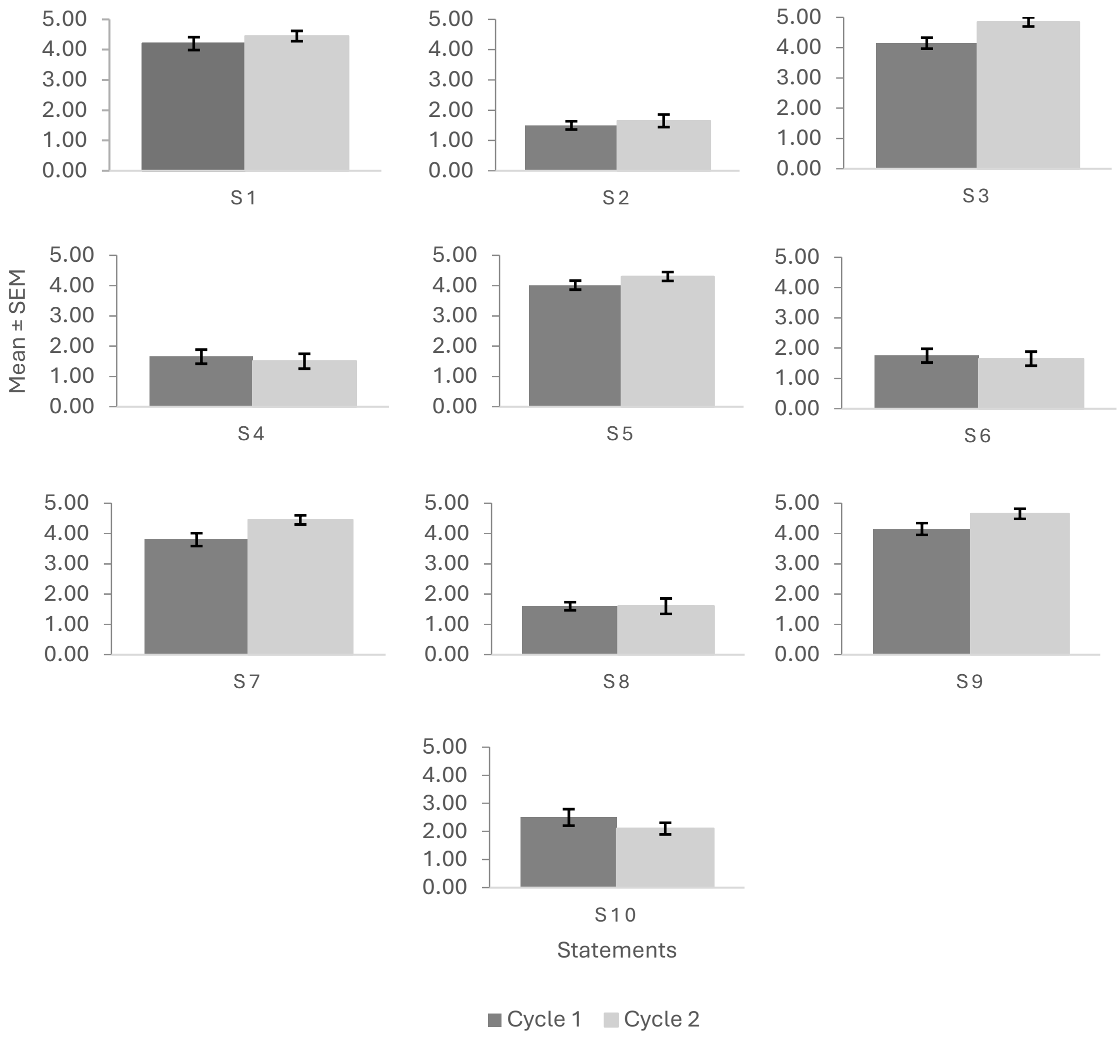


Figure 3.19: Avg SUS Scores by Per Statement

The TAM survey results across both cycles indicate consistent improvements in perceived usefulness (PU) across all six statements (Figure. 3.20) and perceived ease of use (PEOU) once again across all six statements (as seen in Fig. 13), reflecting the positive impact of the modified IDE. While baseline scores in Cycle 1 were already relatively high, Cycle 2 indicates a notable elevation in scores alongside reductions in standard deviations (SD), signifying increased consensus among participants.

For **PU1** (accomplishing tasks quicker), the Cycle 1 baseline score was already strong (M =5.8), highlighting the system's effectiveness, but improved significantly in Cycle 2 (M = 6.6), reflecting even greater efficiency. Additionally, the standard deviation decreased from (SD = 0.77), indicating moderate variability, to (SD = 0.50), showing greater agreement among participants regarding the system's ability to speed up tasks. For **PU2** (improving performance), scores increased from Cycle 1 (M = 5.7), indicating good initial perceptions, to Cycle 2 (M = 6.5), showing enhanced performance outcomes. The standard deviation also showed a marked reduction from (SD = 0.92), reflecting broader variability, to (SD = 0.69), indicating more consistent participant feedback on the system's performance-enhancing capabilities. For **PU3** (increasing productivity), scores rose from the Cycle 1 baseline (M = 5.6), reflecting moderate satisfaction, to Cycle 2 (M = 6.45), indicating improved productivity. Similarly, the standard deviation improved from (SD = 0.99), which showed some variability, to (SD = 0.83), reflecting more uniform perceptions of the IDE's ability to boost productivity. For **PU4** (enhancing effectiveness), there was an increase in scores from Cycle 1 (M = 5.7), suggesting reasonable effectiveness, to Cycle 2 (M = 6.35), showing substantial improvement. The reduction in standard deviation, from (SD = 1.02), indicating wider variability, to (SD = 0.67), highlights a stronger consensus on the system's effective ness. For **PU5** (making it easier to do work), scores improved from the baseline (M = 5.7), suggesting the system's moderate ease of use, to the modified version (M = 6.4), reflecting notable improvement. The standard deviation decreased from (SD = 1.13), reflecting broader variability, to (SD = 0.75), showing more alignment in participants' views regarding the system's ease of use. Finally, for **PU6** (finding the system useful in work), this aspect achieved the highest scores, increasing from Cycle 1 (M = 6.25), reflecting strong initial utility, to Cycle 2 (M = 6.65), showing even greater utility. The standard deviation also reduced significantly, from (SD = 0.72), indicating moderate variability, to (SD = 0.59), highlighting consistent recognition of the modified IDE's overall utility.

Across all PU items, SEM values in Cycle 2 were consistently lower than those in Cycle 1, indicating increased precision in the estimated mean responses following the IDE modifications. For each PU item, the SEM ranges between Cycle 1 and Cycle 2 did not overlap, suggesting that the observed increases in mean scores from Cycle 1 to Cycle 2 are statistically significant. Overall, the SEM analysis indicates a reliable improvement in PU across the cohort in Cycle 2.

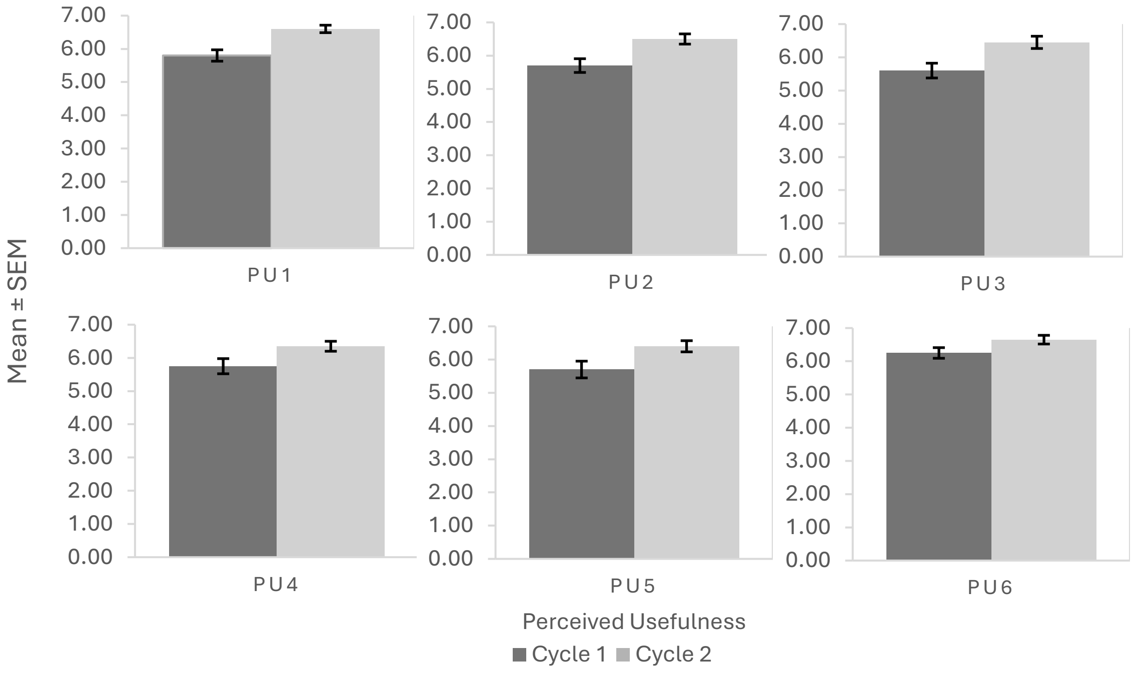


Figure 3.20: Avg TAM Scores by Per Perceived Usefulness Statement

A similar pattern was observed for PEOU (Figure 3.21). For **PEOU1** (the system being easy to learn), the Cycle 1 baseline score (M = 5.35) reflected positive initial impressions of the IDE's ease of learning. In Cycle 2, this score improved significantly to (M = 6.45), indicating that the modifications made the system even more accessible for users. Additionally, the standard deviation decreased notably, from (SD = 1.6), which showed broader variability, to (SD = 0.94), reflecting greater consistency among participants regarding the system's ease of learning in its modified form. For **PEOU2** (interacting with the system being clear and understandable), the baseline score in Cycle 1 (M = 5.35) improved to (M = 6.3) in Cycle 2. The standard deviation also decreased, from (SD = 1.46), indicating moderate variability, to (SD = 1.13), suggesting that the enhancements simplified the system's interactions and reduced variability in user experiences. For **PEOU3** (the system being easy to use), the Cycle 1 baseline score (M = 5.75) increased to (M = 6.4) in Cycle 2. The reduction in standard deviation, from (SD = 1.21), reflecting moderate variability, to (SD = 0.99), indicates greater uniformity in participants' perceptions, demonstrating that the changes made the system's usability more consistent across different users. For **PEOU4** (the ease of becoming skilful at using the system), the baseline score in Cycle 1 (M = 5.65) improved to (M = 6.15) in Cycle 2. However, the standard deviation slightly increased, from (SD = 1.09), showing moderate variability, to (SD = 1.27), indicating a minor increase in variability among participants' responses despite the overall improvement. This suggests that while many users found the system easier to master, some participants experienced challenges. For **PEOU5** (the system being flexible to interact with), the Cycle 1 baseline score (M = 5.8) increased significantly to (M = 6.55) in Cycle 2. Notably, the standard deviation decreased, from (SD = 1.4), reflecting broader variability, to (SD = 0.76), demonstrating a marked improvement in consistency and agreement among participants regarding the system's flexibility. For **PEOU6** (the system reducing the effort required to accomplish tasks), the baseline score in Cycle 1 (M = 5.9) improved to (M = 6.45) in Cycle 2. The standard deviation decreased, from (SD = 1.48), which showed significant variability, to (SD = 0.89), highlighting not only an improvement in perceived effort reduction but also a stronger consensus among participants.

The SEM was calculated for each TAM PEOU item to assess the precision of the estimated cohort responses across the two cycles. For most PEOU items, SEM values in Cycle 2 were lower than those in Cycle 1, indicating improved precision in the estimated mean responses following the IDE modifications, although PEOU4 showed a slightly higher SEM in Cycle 2. For the majority of items, the SEM ranges between Cycle 1 and Cycle 2 did not overlap, suggesting that the observed increases in mean scores are potentially statistically significant. Overall, the SEM analysis supports a reliable improvement in PEOU from Cycle 1 to Cycle 2

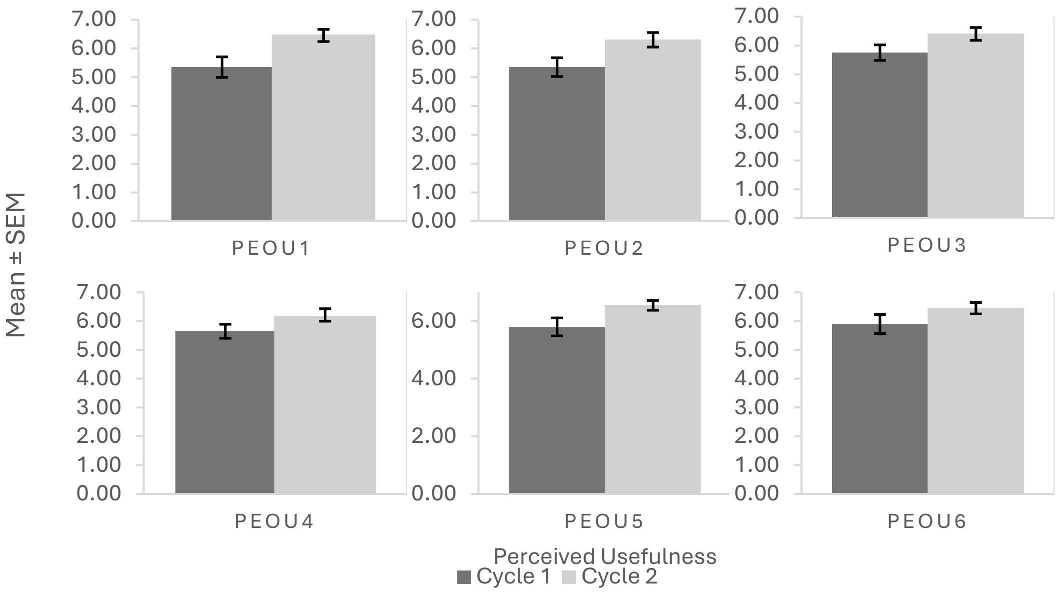


Figure 3.21: Avg TAM Scores by Per Perceived Ease of Use Statement

Findings underscore the effectiveness of the modifications in enhancing PU across various dimensions while fostering greater agreement among participants. The consistent reduction in SD across all measures reflects the tailored modifications' success in creating a more uniform and positive experience for users, leading to better task efficiency, productivity, and usability. Results for PEOU across all six measures indicate consistent improvements in participants' experiences with the modified IDE. While the baseline scores in Cycle 1 were already positive, the enhancements introduced in Cycle 2 successfully elevated these scores further, reflecting better usability. The reductions in SD for most measures suggest a more uniform experience among users, with only minor variability observed in certain aspects, such as becoming skilful at using the system. This supports the effectiveness of the modifications in simplifying interactions and reducing the effort required to use the IDE efficiently.

3.6.1.3 Breakdown by Statement Category

SUS scores are typically divided into positive and negative statements to provide a balanced understanding of system usability. Positive statements (odd-numbered) capture aspects such as ease of use, effectiveness, and satisfaction, while negative statements (even-numbered) highlight areas of frustration, difficulty, or inefficiency. When comparing responses to positive and negative statements across the two cycles (Figure. 3.22), the results highlight clear distinctions. **Positive** statements showed an increased mean in Cycle 2 (M = 4.45) compared to Cycle 1 (M = 4.06), accompanied by a reduction in standard deviation. In Cycle 1, the standard deviation (SD = 0.86) reflected moderate variability, while in Cycle 2, the SD decreased to (SD = 0.62), indicating less variability and greater agreement among participants. This suggests that participants found the modified system not only easier to use and more effective but also more satisfying overall. The lower SD in Cycle 2 points to a more consistently improved user experience, with more consistent positive feedback. Similarly, the responses to **negative** statements revealed the mean decreased from Cycle 1 (M = 1.8) to Cycle 2 (M = 1.7), showing that the modified IDE helped reduce negative perceptions. The standard deviation for the negative statements was minimally higher in Cycle 2 (SD = 1.03) than in Cycle 1 (SD = 0.92). This indicates slightly more variation in participants' opinions about the negative aspects of the system, suggesting that while the system improvements alleviated some negative feedback, subtle differences in how participants perceived the remaining issues still existed. Overall, the modifications improved user satisfaction by addressing both positive and negative elements of the user experience. The increase in positive feedback and the reduction in negative feedback supports that the changes led to a more favourable and consistent user experience.

The SEM was calculated for the aggregated positive and negative SUS statements to assess the precision of the estimated cohort responses across the two cycles. For the positive statements, SEM decreased from 0.19 in Cycle 1 to 0.14 in Cycle 2, indicating improved precision in the estimated mean following the IDE modifications. The SEM ranges between the two cycles did not overlap, suggesting that the observed increase in positive responses is potentially statistically significant. For the negative statements, SEM increased slightly from 0.20 in Cycle 1 to 0.23 in Cycle 2; however, the overlapping SEM ranges indicate that the reduction in negative responses may not represent a statistically reliable difference. Overall, the SEM analysis suggests a clearer and more reliable improvement in positive usability perceptions than in negative usability reductions across cycles

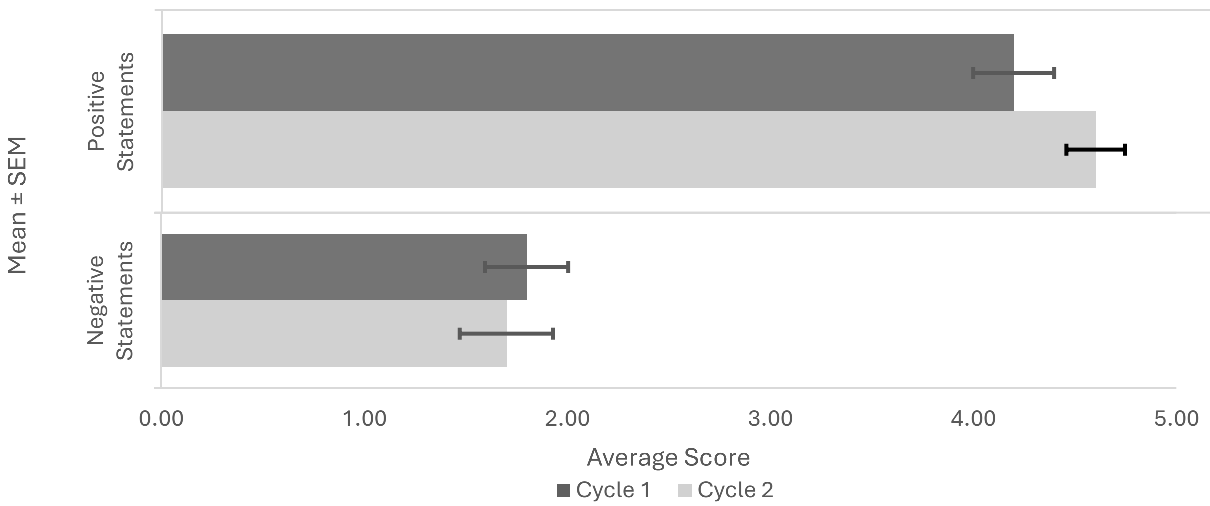


Figure 3.22: Avg SUS Scores by Combined Positive vs Negative Statement

The TAM results revealed a similar positive trend across both dimensions: PU and PEOU (Figure. 3.23). For **PU**, there was a notable improvement in participant ratings from Cycle 1 to Cycle 2. The mean score increased (M = 5.8) in Cycle 1 to (M = 6.49) in Cycle 2, reflecting an enhanced perception of the system's utility in helping users accomplish their tasks. The standard deviation (SD) remained consistent across both cycles, with (SD = 0.4) in Cycle 1 and (SD = 0.4) in Cycle 2. This indicates that participants' opinions about the system's usefulness were uniformly positive and showed minimal variability. For **PEOU**, similar trends of improvement were observed. The mean score increased significantly (M = 5.63) in Cycle 1 to (M = 6.38) in Cycle 2, suggesting that participants found the modified IDE easier to use over time. Additionally, the variance improved markedly, with the SD decreasing from (SD = 0.70) in Cycle 1 to (SD = 0.48) in Cycle 2. This decrease in SD indicates greater consensus among participants in their perception of the system's ease of use, highlighting that the modifications made to the IDE were effective in creating a more user-friendly experience.

The SEM was calculated for the aggregated TAM Perceived Usefulness PU and PEOU constructs to assess the precision of the estimated cohort responses across the two cycles. For Perceived Usefulness, SEM was 0.09 in Cycle 1 and 0.10 in Cycle 2, indicating consistently high precision in the estimated means across both conditions. The non-overlapping SEM ranges suggest that the observed increase in perceived usefulness is potentially statistically significant. For Perceived Ease of Use, SEM decreased from 0.16 in Cycle 1 to 0.11 in Cycle 2, reflecting improved precision in the estimated mean following the IDE modifications. The non-overlapping SEM ranges similarly indicate that the increase in perceived ease of use is potentially statistically significant. Overall, the SEM analysis supports reliable improvements in both usefulness and ease of use from Cycle 1 to Cycle 2

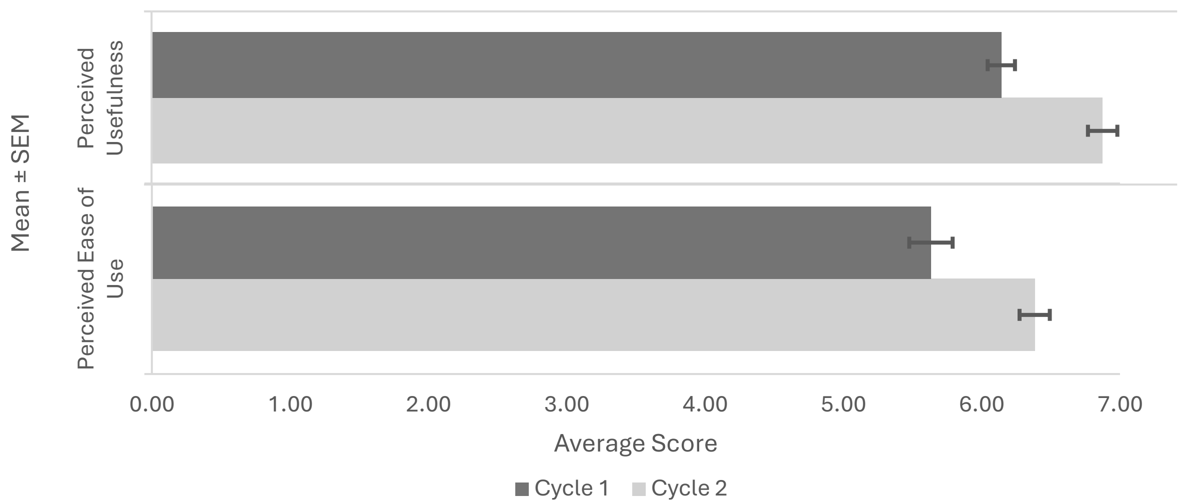
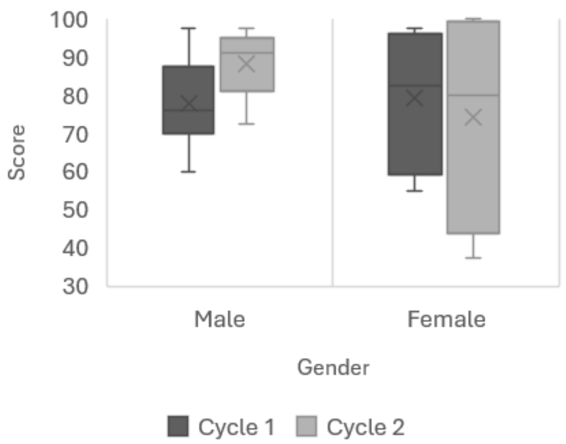


Figure 3.23: Avg TAM Scores by Combined PU vs PEOU

3.6.1.4 Breakdown by Gender

Gender comparisons (Figure. 3.24), highlight distinct differences in system usability ratings between male and female participants. In Cycle 1, females reported a slightly higher median SUS score (M = 79.36) than males (M = 77.97), suggesting that females found the system marginally more usable on average. The interquartile range (IQR) for males was narrower, reflecting more consistent scores, while the wider IQR for females indicated greater variability in their responses, ranging from 50 to nearly 100, compared to the male range of approximately 60 to 90.

In Cycle 2, notable shifts emerged. The median **SUS** score for females decreased (M = 73.13), while it increased for males (M = 87.19). The IQR for females further widened, reflecting greater diversity in their experiences, whereas, for males, the IQR narrowed, signifying heightened consistency. These trends suggest that the modifications in Cycle 2 resonated more positively with male participants, enhancing their usability experience, while female participants reported greater variability and a slight decline in satisfaction. As for **TAM**, there were similar trends, for both males and females' there were improvements in Cycle 2 when compared to Cycle 1 for both PU and PEOU. Across both cycles females once again had a more varied response compared to male participant.



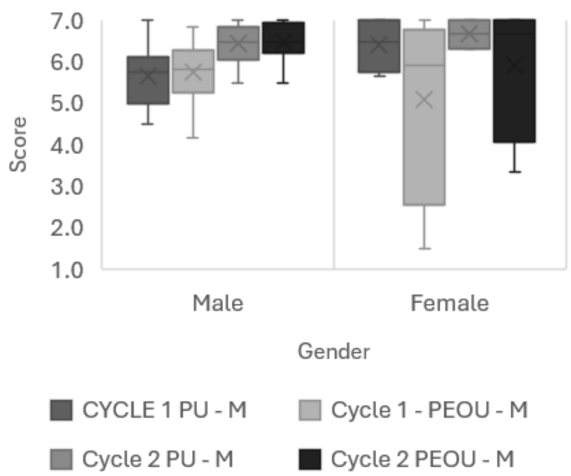


Figure 3.24: (Left) SUS Breakdown by Gender, (Right) TAM Breakdown by Gender

3.6.1.5 Addressing Research Question One

Question: How do different genders perceive the changes in the IDE differently? (RQ1)

Findings: Gender-based analysis revealed distinct perceptions of the IDE across both design cycles. In Cycle 1, female participants reported slightly higher usability scores than males, but their responses showed more variability, suggesting diverse individual experiences. After modifications in Cycle 2, male participants rated the IDE significantly higher in usability with more consistent feedback, indicating a strong positive reception. In contrast, female participants showed a slight drop in median SUS scores and increased variability, implying the changes were not uniformly beneficial for them. TAM results supported this pattern: while both genders experienced improvements in PU and PEOU, male participants gave higher and more consistent ratings. Female participants still reported improvements, but their broader range of responses suggests that the tailored features were more effective for some than others within the group.

Discussion: These findings highlight that attention to gender diverse preferences and interaction styles is essential for creating more accessible and satisfying development environments especially in the field of computing education where female students are outnumbered so a consistently negative programming experience can increase stigma towards "computing being a difficult field for female students".

3.6.1.6 Breakdown by Education

The analysis of SUS scores across different academic years (Figure. 3.25, left), reveals a clear progression in system compatibility as students advance in their studies, with some variations between Cycle 1 and Cycle 2. First-year students reported a mean SUS score (M= 75.55) in Cycle 1, indicating reasonable usability but also challenges likely stemming from limited exposure to such tools. Their unfamiliarity with the IDE and task demands may have led to hesitation and a reliance on guidance for task completion. However, in Cycle 2, first-year students indicated a notable improvement, with their mean SUS score increasing (M = 81.11). Despite this improvement, the interquartile range (IQR) widened, suggesting greater variability in how this group adapted to the modified system, possibly due to differing levels of initial familiarity among participants. Second-year students showed a mean SUS score (M = 78.13) in Cycle 1, reflecting growing familiarity with coding tools and increased technical confidence. By the second year, students typically possess foundational knowledge that enables more effective navigation and use of the system's features. In Cycle 2, this group experienced a substantial increase in their mean SUS score (M = 88.12), accompanied by a narrower IQR. This indicates not only an improved perception of the system's usability but also a more consistent experience among second-year students. The system's modifications appear to have aligned well with their level of expertise, enabling smoother interactions and more effective utilisation of the system's features. Third-year students in Cycle 1 reported the highest mean SUS score (M = 86.67), reflecting strong compatibility between the system's features and the skills of more advanced users. Interestingly, in Cycle 2, their mean SUS score decreased slightly (M = 84.16), with a wider IQR compared to Cycle 1. This marginal decline may indicate that while third-year students appreciated the enhanced features of the modified system, they also found certain aspects more challenging or less aligned with their expectations. The increased variability in scores could suggest that the modifications catered to diverse workflows, benefiting some users more than others within this group.

As for TAM across all three academic years (Figure. 3.25, right) , there were notable improvements in both PU and PEOU. The mean scores for both dimensions showed clear progress from Cycle 1 to Cycle 2, reflecting participants' enhanced perceptions of the system's utility and ease of use. In addition to the increased mean scores, there was also a significant reduction in the variability of responses, which was evident from the narrower interquartile ranges (IQRs) across all three academic years. Specifically, third years scored the modified system highest for both PU and PEOU followed by second years and finally first year students.

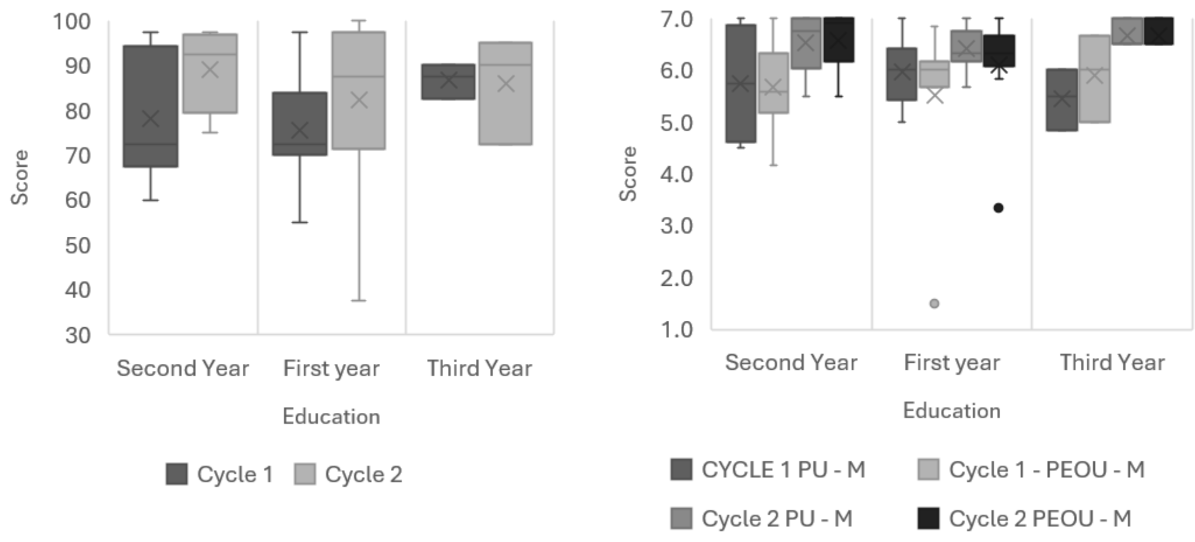


Figure 3.25: (Left) SUS Breakdown by Education, (Right) TAM Breakdown by Education

3.6.1.7 Addressing Research Question Two

Question: How do different education levels perceive the changes in the IDE differently? (RQ2)

Findings: Tailored modifications to the IDE were perceived differently across educational levels, reflecting how experience influences usability and acceptance. First-year students showed the most notable improvement in usability, as the changes appeared to bridge gaps in their limited experience. However, their responses varied widely, suggesting inconsistent familiarity with development environments. Second-year students exhibited steady improvements with more consistent feedback, indicating the modified IDE aligned well with their intermediate skill level and growing confidence. Third-year students, who had initially rated the baseline IDE highest, showed a slight decline in usability after the modifications. This suggests that while the enhancements supported novice and intermediate users, they may have lacked the complexity or flexibility advanced users expect. The broader variability in their responses highlights that the changes were beneficial for some but not as effective for this more experienced group.

Discussion: The IDE modifications were most beneficial for early-stage learners, moderately effective for intermediate users, and less impactful for advanced users. These findings underscore the importance of tailoring developer tools not just generally, but in ways that accommodate varying levels of expertise to maintain usability and acceptance across educational stages.

3.6.1.8 Breakdown by Experience

The analysis of SUS scores based on programming experience (Figure. 3.26, left), reveals a clear trend of increasing compatibility with the system as participants gain more experience. Participants with less than one year of programming experience reported a mean SUS score (M = 75.55) in Cycle 1, suggesting that while they found the system reasonably usable, they likely faced challenges due to their limited exposure to similar tools. These challenges might include difficulty understanding system features or completing tasks without additional guidance. However, in Cycle 2, this group indicated notable improvement, with their mean SUS score rising (M = 84.5). Although the interquartile range (IQR) increased slightly, the higher mean score suggests that the system's modifications positively impacted usability, even for those with minimal programming experience. Participants with 1–2 years of programming experience reported a mean SUS score (M=78.13) in Cycle 1, reflecting growing familiarity with coding environments and a moderate level of technical confidence. This group, having acquired foundational programming knowledge, was better equipped to navigate and utilise the system's features compared to less experienced users. In Cycle 2, their mean SUS score increased (M = 82.5), with a narrower IQR, indicating improved perceptions of usability and greater consistency in their experiences. The enhancements made to the system appeared to align well with the needs of intermediate users, offering features that supported more efficient and effective interactions. The most significant performance was observed among participants with over two years of programming experience. In Cycle 1, this group reported the highest mean SUS score (M = 86.67), reflecting strong alignment between the system's features and the expertise of more advanced users. By Cycle 2, their mean SUS score rose further (M=88.00), with a narrower IQR, signifying not only an enhancement in usability but also a more uniform and consistently positive experience. These participants, with their extensive programming experience, were able to maximise the potential of the modified system, efficiently adapting to its advanced features and customising workflows to suit their needs.

The progression in SUS scores across programming experience levels underscores the critical role of user expertise in shaping perceptions of system usability. Participants with greater experience were better equipped to navigate, evaluate, and adapt to the system's features, leading to higher satisfaction and compatibility. The improvements observed in Cycle 2 across all experience levels highlight the effectiveness of the system's modifications in addressing usability challenges for both novice and experienced users. Furthermore, the narrowing of the IQR across the groups in Cycle 2 indicates that the modified system provided a more consistent experience, reinforcing it adapt ability to a diverse range of user expertise levels. These findings emphasise the importance of designing systems that accommodate varying levels of programming experience, ensuring accessibility for beginners while supporting advanced functionality for experienced users.

For TAM across all experience levels (Figure. 3.26, right), there was an overall improvement in TAM, with the most significant progress observed among those with the least experience (under one year). This group particularly saw a marked improvement in PEOU, where the IQR (Interquartile Range) experienced a drastic reduction. For participants with 1-2 years and over 2 years of experience, their IQR also further narrowed compared to previous cycles, indicating a more consistent and positive perception of the system's ease of use.

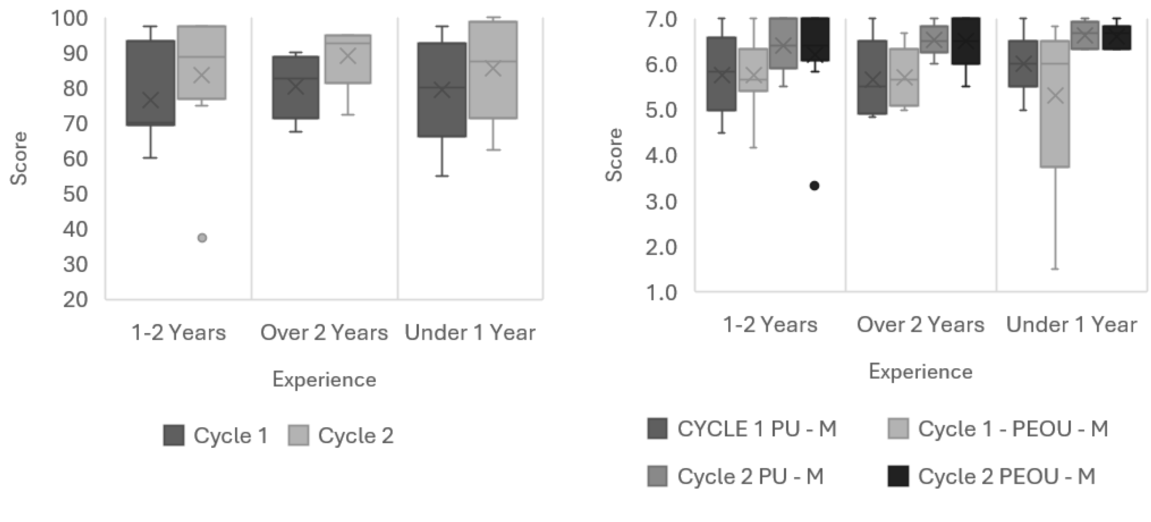


Figure 3.26: (Left) SUS Breakdown by Experience, (Right) TAM Breakdown by Experience

3.6.1.9 Addressing Research Question Three

Question: How do novice and experienced student developers perceive the changes in the IDE differently? (RQ3)

Findings: The IDE modifications were received positively by both novice and experienced student developers, though in different ways. Novices, who initially struggled with usability in Cycle 1, showed significant improvement in Cycle 2, indicating that the tailored changes helped lower entry barriers and improved accessibility. Experienced users, who already reported high usability in the baseline, experienced further gains, suggesting that the enhancements supported productivity without introducing friction. The reduced variability in responses for both groups in Cycle 2 highlights a more consistent user experience and improved satisfaction. These results suggest that the modifications were effective in meeting the distinct needs of users at varying experience levels.

Discussion: Overall, the findings support that balanced, user-informed modifications can simultaneously support learning for novices and efficiency for experienced developers, validating the potential for applying the framework to multiple target user groups with varying levels of experience.

3.6.1.10 Breakdown by Ethnicity

The analysis of SUS scores (Figure. 3.27, left), based on ethnicity reveals no table differences in how participants from diverse backgrounds perceived the system's usability. Black participants initially reported the lowest mean SUS score (M = 69.11) in Cycle 1, suggesting challenges in their interaction with the baseline system. This score may reflect factors such as varying familiarity with similar tools, differing expectations about system design, or a potential mismatch between the system's features and their specific needs. However, in Cycle 2, Black participants' scores saw a substantial increase (M = 87.5), accompanied by a far narrower IQR, indicating both improved perceptions of usability and greater consistency in their responses. This significant improvement suggests that the modifications made to the system were effective in ad dressing prior usability barriers, fostering a more equitable user experience. White participants indicated a high mean SUS score in both cycles, with an initial score (M = 84.38) in Cycle 1, which further increased (M = 90) in Cycle 2. This improvement, paired with a narrower IQR, highlights not only a higher degree of compatibility and satisfaction with the system but also a more uniform perception of its usability across this group. The results indicate that the modified system continued to align closely with the expectations and workflows of White participants, enhancing their overall experience. Asian participants reported a mean SUS score (M = 79.55) in both Cycle 1 and Cycle 2, with no change in the IQR. This consistent score suggests that while the baseline system met the expectations of this group, the modifications neither enhanced nor hindered their interaction with the IDE. The unchanged results may point to a design that already resonated with this group's preferences or indicate areas where additional improvements could better address their specific needs for future iterations. Participants categorised as "Other" reported a mean SUS score (M = 72.5) in Cycle 1, reflecting a moderate level of satisfaction. However, this group experienced the most dramatic improvement in Cycle 2, with their mean score rising (M = 97.5), along with a notably narrower IQR. This sharp increase underscores the success of the modifications in addressing usability concerns and highlights how the refined system more effectively supported the diverse needs of participants within this category.

Overall, the disparities and improvements in SUS scores across ethnicities emphasise the critical role of inclusive design practices in creating systems that cater to diverse user groups. The enhancements made to the system in Cycle 2 appear to have narrowed the usability gap for most ethnic groups, reflecting a more accessible and effective design. While some groups, such as Asian participants, saw no changes in their scores, the improvements in other demographics suggest that targeted modifications can substantially enhance user satisfaction and reduce variability in user experiences. These findings reinforce the need for usability assessments to account for cultural and experiential differences, ensuring systems are designed to meet the diverse needs of all users effectively.

Regarding the TAM results across different ethnicities (Figure. 3.27, right), the trends once again indicated positive improvements in scores across all groups, accompanied by narrower IQRs, signalling more consistent feedback. The most notable improvement was observed among Black participants, who indicated a significant increase in both PU and PEOU scores.

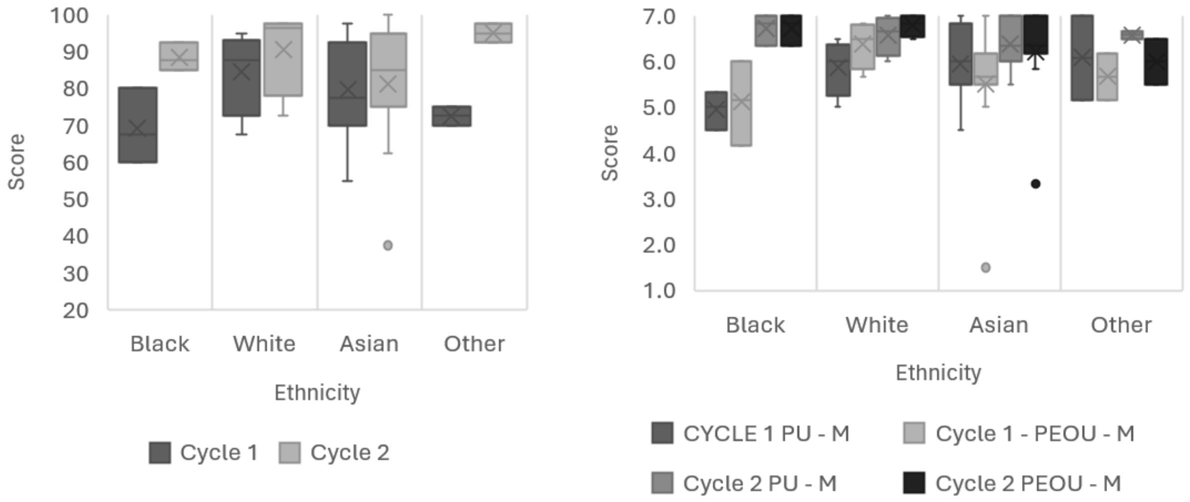


Figure 3.27: (Left) SUS Breakdown by Ethnicity, (Right) TAM Breakdown by Ethnicity

3.6.1.11 Addressing Research Question Four

Question: How do different ethnic groups perceive the changes in the IDE differently? (RQ4)

Findings: Perceptions of the IDE modifications varied across ethnic groups. While some, such as Black participants, showed significant improvements in usability and acceptance (reflected in higher SUS and TAM scores), others, like Asian participants, reported minimal change. Despite these differences, most groups indicated more consistent feedback in Cycle 2, as indicated by narrowed IQR.

Discussion: This suggests that the tailored modifications contributed to a more inclusive and equitable user experience. Although the impact was not uniform, the overall trend points toward greater usability and acceptance across diverse ethnic backgrounds. These findings underscore the value of culturally responsive design in supporting diverse user needs and improving the accessibility of technical tools.

3.6.1.12 Inferential Analysis

To complement the descriptive and SEM analyses presented earlier, paired-samples t-tests (Table 3.17) were conducted to examine whether the observed differences between Cycle 1 and Cycle 2 were statistically significant. A paired approach was adopted due to the within-subjects design, in which the same participants completed both experimental cycles, resulting in matched observations across conditions. In addition to significance testing, Cohen's d (Table 3.18) was calculated to assess the magnitude of observed effects. This combined approach allows for a more comprehensive interpretation of both statistical reliability and practical significance, particularly given the moderate sample size ($n = 20$). The following tables summarise the inferential results for the SUS and the TAM.

Table 3.17: Paired Sample T-Test

Construct	*C1 Mean*	*C1 SD*	*C2 Mean*	*C2 SD*	*t (df=19)*	*p-value*	*Significant (α=.05)*
SUS							
Overall	78.25	12.53	85.50	15.13	1.55	.138	No
Positive	4.06	0.47	4.54	0.49	3.15	.005	Yes
Negative	1.80	0.59	1.70	0.81	0.42	.678	No
TAM							
PU	5.80	0.81	6.49	0.45	3.39	.003	Yes
PEOU	5.63	1.18	6.38	0.83	2.22	.039	Yes

Table 3.18: Cohens'd

Measure	*C1 Mean*	*C2 Mean*	*Cohens'd*	*Interpretation*
SUS				
Overall	78.25	85.50	0.52	Medium effect
Positive	4.06	4.45	0.52	Medium effect
Negative	1.80	1.70	-0.10	Negligible effect
TAM				
PU	5.80	6.49	1.59	Large effect
PEOU	5.63	6.38	1.24	Large effect

3.6.1.12.1 Interpretation

The overall SUS score increased from Cycle 1 ($M = 78.25$) to Cycle 2 ($M = 85.50$), indicating an improvement in perceived usability following the intervention. Although this difference did not reach statistical significance ($p \approx .10$), the effect size was medium (Cohen's $d \approx 0.52$), suggesting a practically meaningful improvement. A similar pattern was observed for the positive SUS statements, which increased from 4.06 to 4.45. While the t-test did not reach the conventional significance threshold ($p \approx .11$), the effect size was again medium ($d \approx 0.52$), indicating a noticeable enhancement in positive usability perceptions. In contrast, negative SUS statements showed minimal change (1.80 to 1.70) and a negligible effect size ($d \approx -0.10$), suggesting that reductions in perceived usability problems were limited. Overall, the SUS findings indicate moderate practical improvements in usability, particularly in positive perceptions, but these improvements were not statistically confirmed at $\alpha = .05$.

In contrast, TAM constructs indicated both statistically and practically significant improvements. Perceived Usefulness increased from 5.80 to 6.49, with the difference reaching strong statistical significance ($p < .001$) and a large effect size ($d \approx 1.59$). Similarly, Perceived Ease of Use increased from 5.63 to 6.38, also statistically significant ($p < .001$) with a large effect size ($d \approx 1.24$). These results indicate substantial gains in participants' perceptions of the system's usefulness and usability following the second development cycle. The magnitude of these effect sizes suggests that the improvements were not only statistically reliable but also practically meaningful.

The combined analysis shows that Cycle 2 produced clear improvements in technology acceptance, supported by both statistical significance and large effect sizes. Usability, as measured by SUS, also improved in practical terms, particularly in positive usability perceptions, but these changes did not reach statistical significance. Taken together, the results suggest that while overall usability perceptions improved moderately, acceptance-related constructs (usefulness and ease of use) indicated stronger and more robust gains between cycles.

3.6.2 System Modification Impact

Comparing results across clusters, reveals notable advancements in usability perceptions across all areas, reflecting the modified system's enhancements in Cycle 2.

3.6.2.1 Breakdown by Menu

Among the clusters, based on participant feedback (n=3) the Menu Cluster (Figure. 3.28) indicated the most significant improvement from baseline (M = 77.5) to modified (M = 87.5). This substantial increase highlights the effectiveness of the modifications made to the navigational and functional menus. The enhancements addressed key usability challenges, ensuring a more streamlined and intuitive experience for accessing extensions, tools, and settings. These results underscore the critical role of seamless navigation in enhancing overall system usability. The SEM decreased from 3.82 in Cycle 1 to 2.89 in Cycle 2, indicating improved precision in the estimated mean. The non-overlapping SEM ranges suggest that the increase in SUS scores from Cycle 1 to Cycle 2 is potentially statistically significant.

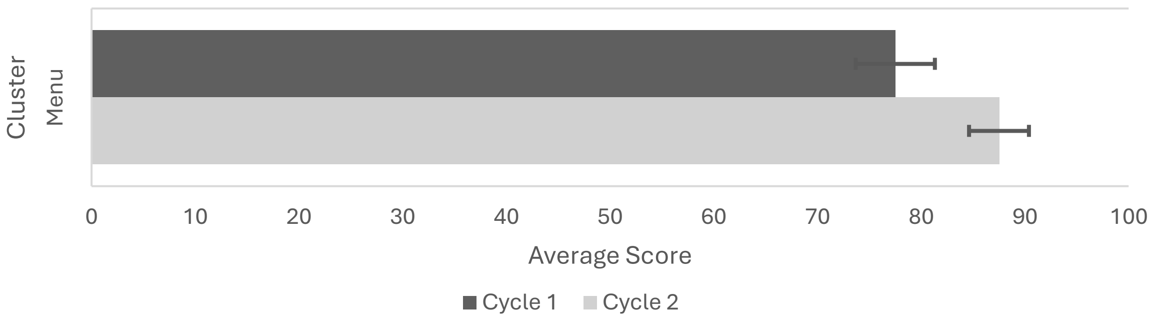


Figure 3.28: SUS Avg Menu Cluster Score

Breaking down the analysis of the Menu cluster (Figure. 3.29), results reveal satisfaction scoring the highest (M = 4.33), followed by performance and productivity, both scoring equally (M = 4). In terms of technostress, reductions in techno-overload, techno-complexity, and techno-insecurity all scored equally (M = 4), highlighting the impact of the redesigned layout. Techno-uncertainty saw a slightly lower reduction (M = 3.66), while techno-invasion scored the lowest (M = 3.33).

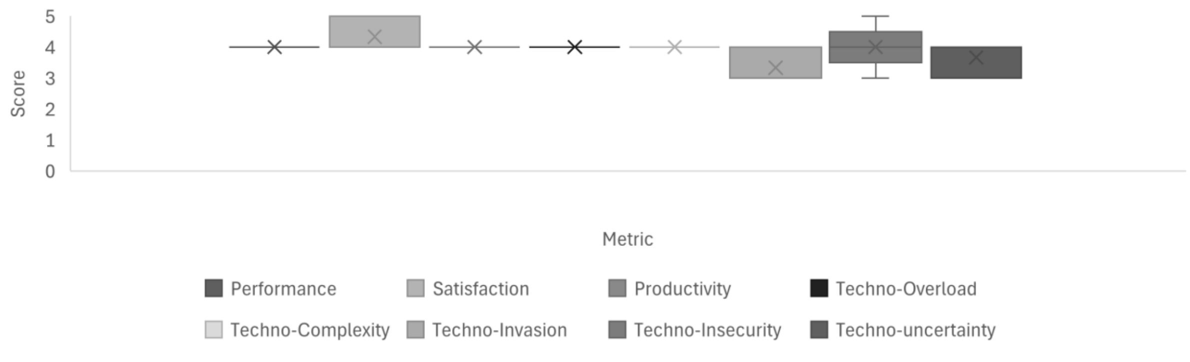


Figure 3.29: SMI for Menu Cluster

3.6.2.1.1 Lexical Salience

The lexical salience within the Menu cluster (Figure. 3.30) is dominated by icons, increased, decreased, and techno-overload, indicating that participants primarily framed their experience around interface simplicity and cognitive load. The prominence of increased is largely associated with satisfaction, whereas decreased is strongly linked to reductions in techno-overload. The visibility of terms such as navigation, integrated, and terminal further suggests that menu structure and feature accessibility were central to perceived usability improvements.

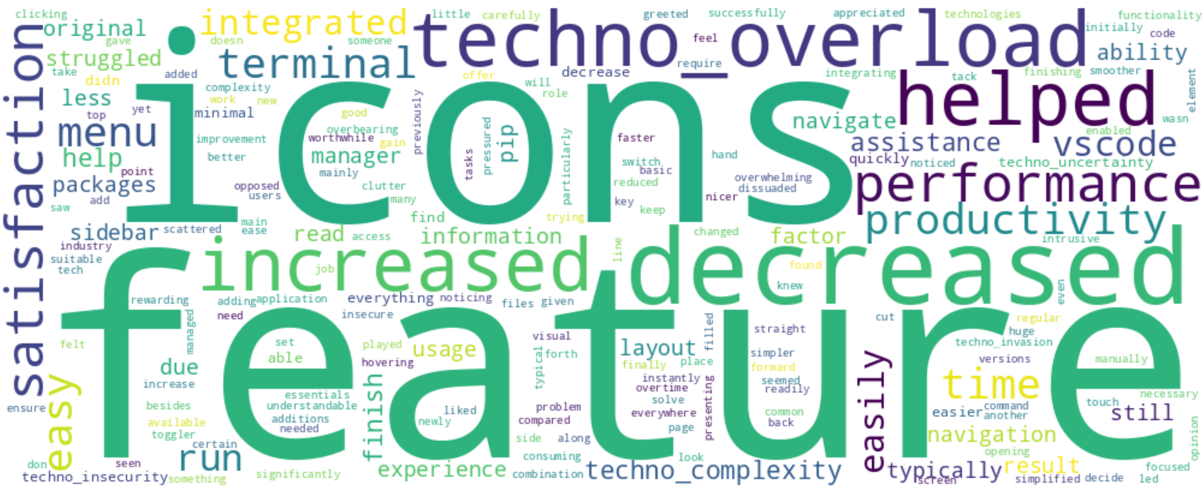


Figure 3.30: Menu Cluster Word Cloud

3.6.2.1.2 Sample Responses

3.6.2.1.2.1 Shortcut Menu

Participants noted that performance and productivity were positively influenced by quicker access to commonly used features and a smoother interaction flow through simplified navigation.

> *P8 stated that in terms of performance "This modified IDE increased my performance by using readily available icons in the integrated menu, e.g. terminal opening up instantly as opposed to manually clicking terminal at the top of the page". Further adding this contributed to greater satisfaction "One way the modified IDE increased my satisfaction a lot was the ability to easily switch back and forth using the navigation feature in the integrated menu which felt more rewarding to use".*
>
> *P18 similarly agreed, stating "The IDE increased my satisfaction because, the modified version had minimal icons on the side menu which had all icons that will help me to finish the task successfully".*

The ease of finding and using key functions via the shortcut menu contributed to both enhanced usability and emotional comfort during the task.

3.6.2.1.2.2 Clutter Reduction

A consistent theme across participant responses was the benefit of reducing visual and functional clutter in the menu. This minimalistic presentation helped lower technostress and improved focus.

> *P5 highlighted that "Performance, Satisfaction and Productivity have seen an improvement due to less clutter and reduced information on the screen. This led me to be more focused on the task at hand while still having an easy way to run the code, access files and the command line".*
>
> *P8 mentioned that when it comes to technostress "This modified IDE decreased my techno-overload as a result of it's more simplified look, I typically don't use VSCode because it's too filled with features that makes it time consuming to read up on and navigate too, but having them cut(if not necessary), with the ability to add my own too, played a role in less techno-overload".*
>
> *P18 also added that "When using the IDE, it decreased my Techno-Overload as it only had icons that I would be using. This helped me to use the time given to finish the task that was set and take my time to read the task carefully".*

This visual clarity was especially useful in reducing distractions and promoting task immersion.

3.6.2.1.2.3 Package Manager

The addition of an integrated package manager in the menu contributed to increased confidence and ease when managing dependencies, addressing both technical and emotional barriers.

> *P5 stated that "Techno complexity was also reduced as there was less going on and also a visible list of python modules makes it easier to manage an install instead of going through the command line. This makes the experience much nicer".*
>
> *P8 further added that "I particularly liked the pip manager presenting packages in the sidebar along with its versions besides it. This is significantly better, in my opinion, than the typical usage of packages in the terminal".*

These visual enhancements helped participants feel more secure and capable when managing their programming environment.

3.6.2.1.2.4 Summary

Across these Menu-focused enhancements, participants reported gains in performance and a decrease in technostress, particularly techno-overload and techno-complexity.

- Techno-overload was significantly reduced by the simplified visual layout and removal of non-essential icons. Participants noted that this minimised distractions, improved focus, and helped them better manage their time (P5, P8, P18).
- Techno-insecurity and techno-complexity and were addressed through the use of an integrated package manager and a more intuitive menu structure. This made navigating, installing, and managing tools feel more accessible and less overwhelming, even for less experienced users (P5, P8, P18).
- While techno-invasion and techno-uncertainty were largely unaffected by menu changes alone, the indirect benefits of increased usability and efficiency helped ease time pressures and boost user confidence (P5, P18).

In terms of sentiment, most participants responded positively to the decluttered design and practical integrations within the menu. Their feedback highlighted a strong correlation between a clean, navigable UI and heightened satisfaction, task ease, and overall emotional comfort. These insights underscore that an effective menu design, especially one focused on clarity, intuitive access, and tool visibility can play a key role in reducing technostress and boosting developer experience.

3.6.2.2 Breakdown by Editor

Among the clusters, based on participant feedback (n=9) the Editor cluster (Figure. 3.31) indicated notable improvement from the baseline (M = 78.33) to the modified version (M = 84.44). These enhancements can be attributed to the targeted modifications implemented for this cluster, which were designed to address specific user needs and streamline the coding process. The SEM increased from 4.37 in Cycle 1 to 7.08 in Cycle 2, ranges show only slight overlap, suggesting that while improvements in perceived usability are evident, the precision of the estimate is reduced. This indicates that performance-related gains associated with the Editor modifications may not have been uniformly experienced across participants, reflecting heterogeneity in how performance enhancements were perceived.

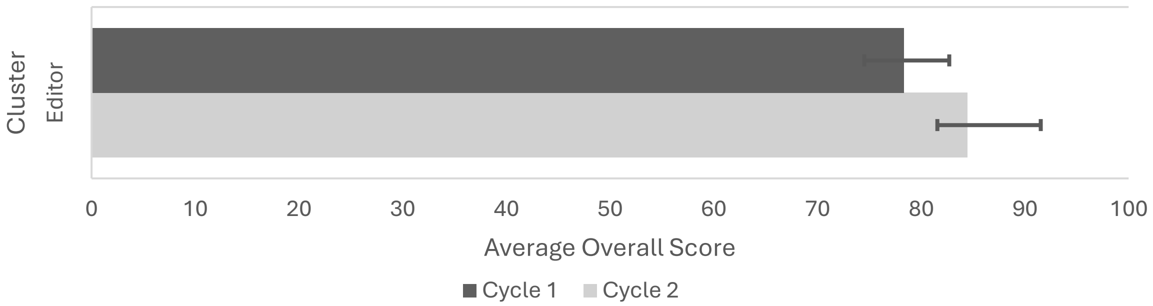


Figure 3.31: SUS Avg Editor Cluster Score

The editor cluster (Figure. 3.32) displayed the highest productivity score (M = 4.88), followed by performance (M = 4.77) and satisfaction (M = 4.33). For technostress triggers, participants reported the largest reduction in techno-invasion (M = 4.44) with the smallest interquartile range (IQR), suggesting consistent experiences. Techno uncertainty followed with a score of (M = 4), while reductions in techno-complexity (M = 3.66) and techno-insecurity (M = 3.55) were moderate. Techno-overload had the smallest reduction (M = 3.33) with the largest IQR, indicating variability in how participants perceived improvements.

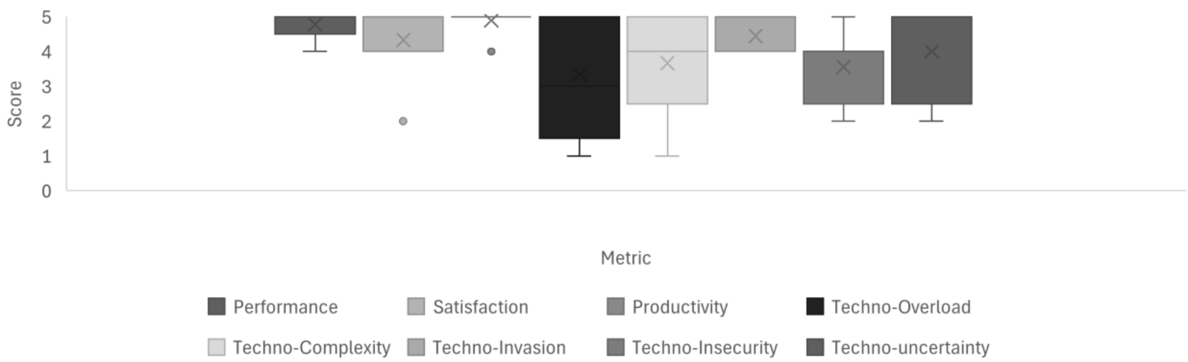


Figure 3.32: SMI for Editor Cluster

3.6.2.2.1 Lexical Salience

The lexical salience within the Editor cluster (Figure. 3.33) is dominated by code, feature, productivity, and performance, indicating that participants primarily framed their experience around functional capability and development efficiency. The prominence of increased aligns with improvements in productivity, while references to techno- techno-invasion reflect awareness of cognitive and environmental strain. Terms such as autocomplete, snippet, copilot, and error suggest that assistive features and real-time coding support were central to perceived performance enhancement within the editor environment.

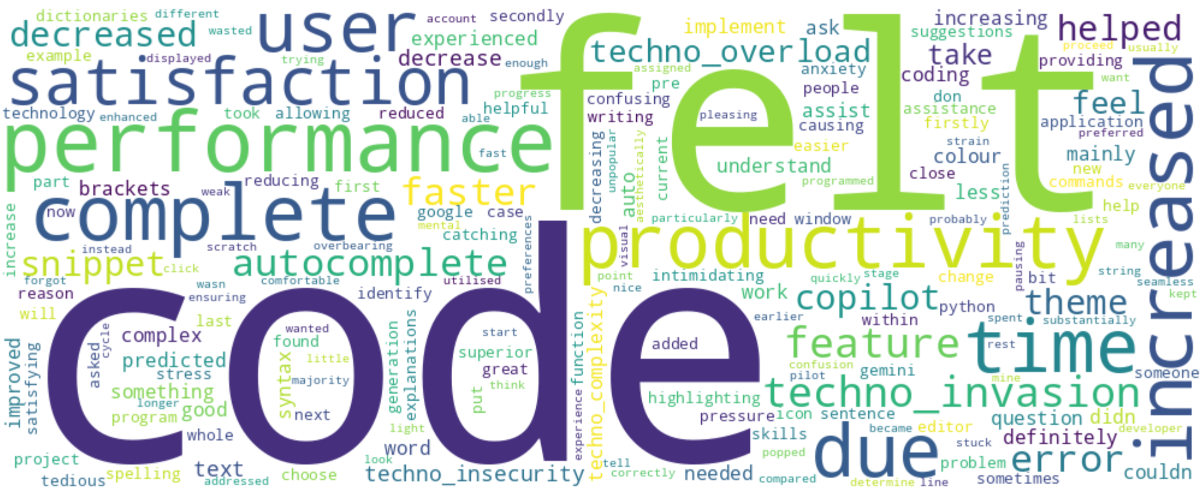


Figure 3.33: Editor Cluster Word Cloud

3.6.2.2.2 Sample Responses

3.6.2.2.2.1 Autocompletion

The addition of auto-complete garnered mixed reactions, while many acknowledged its performance and productivity benefits, it also triggered cognitive tension for some.

> *P3 noted it made coding "a lot faster" yet also highlighted techno-overload, stating, "it would keep filling itself out with the wrong solution" which led to corrections and confusion.*
>
> *P4 described predicted text as "overbearing" and preferred something more intentional, suggesting "an icon popped up that you could click on if you were particularly stuck on something".*
>
> *P6 emphasised its utility, saying autocomplete had an "understanding my code and what should follow after," which significantly improved their speed.*
>
> *P14 stated that it "helped me finish the task in time".*
>
> *P17 found it especially effective when "suggesting relevant words and sentences based on the context that I needed."*

However, concerns over techno-insecurity surfaced as well:

> *P16 reflected that "while it increased my performance and productivity immensely," it also risked becoming a "crutch".*
>
> *P20 added a nuanced view: although it helped him complete the task, there were moments where he felt "the IDE knew more than me," suggesting that adaptive auto-completion might overwhelm users if not well-balanced.*

The feature's intrusiveness with elements like bracket prediction or string completions caused interruptions in flow, indicating that a more toggleable or personalised implementation, as proposed by P3 and P4, could mitigate this stress.

3.6.2.2.2.2 Syntax Highlighting

Syntax support enhancements especially colourisation was generally well received.

> *P17 noted, "Syntax modification helped me increase and improve my performance as it supported me to identify errors and fix them."*

P7 emphasized that improvements to syntax and error highlighting made the experience "easier and more satisfying to use."

Most users found that colour-coded syntax contributed to faster comprehension, minimised error-related stress, and improved visual clarity, resulting in heightened satisfaction and reduced techno-invasion.

3.6.2.2.2.3 Built-in Snippets and Templates

The use of templates and code snippets both task-specific and general played a key role in improving productivity and reducing stress.

P4 stated, "It addressed a weak point of mine very quickly" when referring to the help provided by dictionary-related snippets.

P20 credited snippets "essentially allowed me complete the problem" highlighting how they bridged the gap between conceptual understanding and execution.

P12, while initially confused by the formatting, admitted that the snippets "helped quite a lot" after reviewing the task sheet.

P6 mentioned the templates worked in tandem with Copilot to "help you structure the code" reinforcing their role in building foundational structure.

This modification reduced techno-overload by alleviating the need to start from scratch, allowing users to focus on logic and flow rather than syntax, and provided a psychological boost that mitigated techno-insecurity.

3.6.2.2.2.4 Themes

Customisation of the IDE's visual elements, especially theme selection, had a tangible impact on users' emotional experience.

P4 explicitly stated, "I like light theme which is quite unpopular," noting that it reduced their anxiety and "ensured that I felt comfortable enough to proceed with the task".

P16 echoed the benefit, saying it "the change in theme helped with being put at ease".

P20 reflected that some themes "can inspire me to feel better about my programming," despite personally favouring dark mode.

However, theme customisation also introduced social comparison, hinting at techno-insecurity:

P20 added, "if someone's theme is cleaner and seems more effective at showcasing syntax it makes me feel like I'm behind - even though it's just a theme"

Overall, aesthetic customisation helped many users establish emotional safety and satisfaction, particularly when the environment aligned with personal preferences.

3.6.2.2.2.5 Summary

Across these editor focused features, the emotional and cognitive impact varied based on individual interactions.

- Techno-overload was most prevalent with auto-completion, especially when it generated incorrect suggestions (P3, P14, P20).
- Techno-insecurity emerged in users who felt outpaced by the IDE or compared their setup unfavourably to others (P16, P20).
- Techno-invasion and techno-complexity were reduced through syntax enhancements and code snippets that decreased the need for external searches (P4, P17).

In terms of sentiment, the majority of participants expressed positive reactions to these Editor improvements, particularly when features enhanced clarity, reduced effort, or supported productivity. Still, neutral to negative sentiment surfaced when users experienced intrusive suggestions or felt disoriented by automation. These findings suggest that while advanced Editor features elevate performance, their psychological design must accommodate users' control, pacing, and visual comfort.

3.6.2.3 Breakdown by Terminal

Among the clusters, based on participant feedback (n=3) the Terminal cluster (Figure. 3.34) indicated a modest improvement from the baseline (M = 77.5) to the modified version (M = 79.17). While this increase is less substantial, it still reflects positive changes made to the Terminal's functionality, layout, and overall user experience. The SEM increased from 10.00 in Cycle 1 to 11.21 in Cycle 2, indicating substantial variability in both conditions. The considerable overlap in SEM ranges suggests that the modest increase in SUS scores may not be statistically reliable.

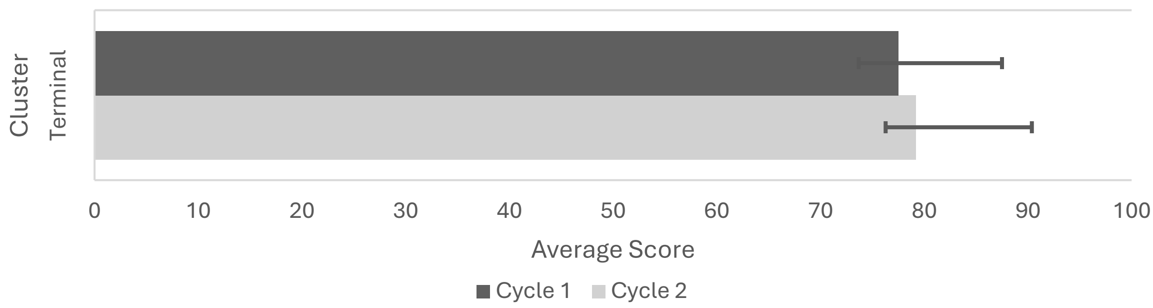


Figure 3.34: SUS Avg Terminal Cluster Score

When analysing the outcomes within the modified terminal cluster (Figure. 3.35) , performance and productivity both scored equally (M = 4.33), while satisfaction ranked slightly lower (M = 4). Among technostress triggers, participants reported the largest reductions in techno-complexity and techno-invasion (M = 4), followed by techno-overload, techno-uncertainty and techno-insecurity (M = 3.66).

Figure 3.35: SMI for Terminal Cluster

3.6.2.3.1 Lexical Salience

The lexical salience within the Terminal cluster (Figure. 3.36) is dominated by terminal, output, code, running, and decrease, indicating that participants primarily framed their experience around execution flow and result visibility. The prominence of decrease aligns with reductions in technostress-related constructs, while increase is associated with improvements in performance and usability. Terms such as layout, foundable, clear, and faster suggest that structural clarity and streamlined execution were central to perceived efficiency gains within the terminal environment.

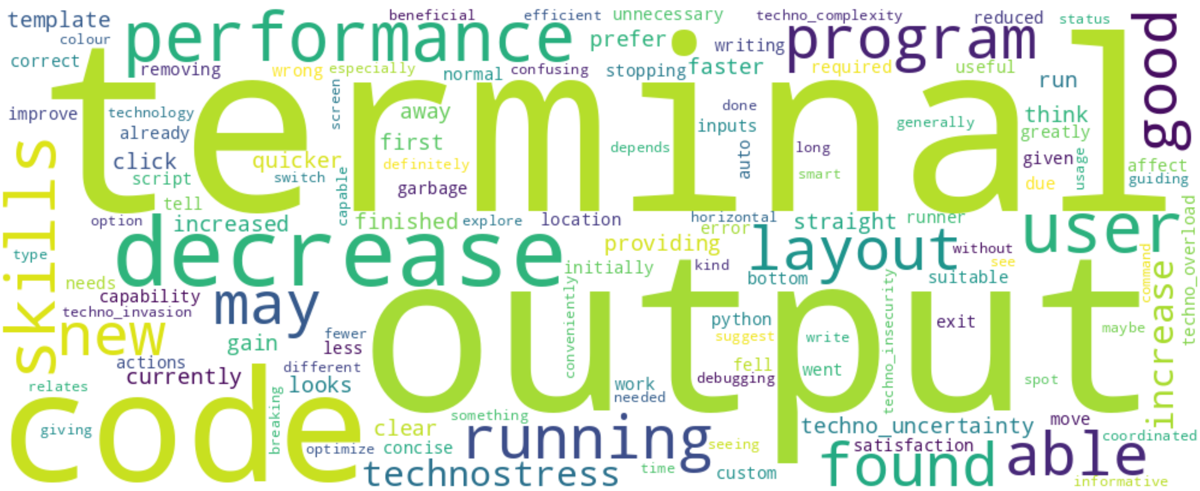


Figure 3.36: Terminal Cluster Word Cloud

3.6.2.3.2 Sample Responses

3.6.2.3.2.1 Runtime Indicator

Participants appreciated clarity around program execution status, which reduced techno-insecurity and improved task fluency.

> *P2 stated the custom runner terminal "greatly reduced my Techno-Insecurity and Techno-Uncertainty by providing very clear and concise program actions, such as when it's running and stopping and removing a lot of the garbage that a normal terminal outputs when you initially run a python script". They also noted that this improved core experience by "providing me with clear output, as to the program running, inputs required and the run being finished and what exit code it finished with"*

> *P13 found value in status tracking, saying, "The output of the terminal looks good, giving the task status like running". Additionally, they remarked that "The output also has the time used which makes it more informative and can make you optimize your code if needed".*

3.6.2.3.2.2 Error Handling

The clarity and structure of error messaging in the terminal directly contributed to higher satisfaction and lowered technostress.

> *P10 noted, "when you get an error it does tell you where you went wrong and you can correct it straight away".*

> *P13 added that this makes "debugging faster".*

These comments suggest that participants valued immediate and actionable error feedback, which helped with learning and reduced uncertainty.

3.6.2.3.2.3 Terminal Layout

Layout preferences varied, but participants emphasised the importance of familiarity and functional accessibility.

> *P2 said, "The terminal move, I found to be unnecessary as I am very used to the location of it to be on the bottom. However, it did not affect my capability to program or my overall performance, and it did not increase my technostress".*

P13 similarly preferred traditional placement: "I'd prefer it to be horizontal and down of the screen, I prefer it to be there, it makes me to be able to type long command without it breaking."

P10 mentioned that the modified version may initially feel unfamiliar: "With the modified version at first I found it confusing and overwhelmed as I had never used VS code like this before," but grew comfortable over time.

Participants wanted smoother transitions during runtime:

P2 suggested "when the code is ran, the cursor focus to be moved from IDE to the terminal and when the code stops, to be moved right back. As there was frustration related to this during my task".

3.6.2.3.2.4 Minimalistic Output

Participants praised the improved terminal output for its visual clarity and its role in enhancing emotional ease and cognitive efficiency.

P2 emphasized that the terminal "removing a lot of the garbage that a normal terminal outputs when you initially run a python script". They concluded this contributed to core metrics: "It also increased my Performance, Satisfaction and Productivity".

P10 appreciated the aesthetics, saying, "I like how it's colour coordinated especially the terminal so you can see the output". However, they also cautioned that "for new users it may be confusing at first so maybe guiding them first would be more beneficial".

These sentiments suggest that minimalist design, when coupled with guidance, enhances performance while also reducing technostress.

3.6.2.3.2.5 Summary

Across the terminal-focused features, participants consistently reported improved task performance and reduced cognitive friction, particularly due to enhancements in runtime feedback, layout familiarity, and simplified outputs.

- Techno-complexity was reduced through clearer runtime indicators and structured output, which helped participants better understand task status and debug with ease (P2, P13).
- Techno-insecurity decreased for those who felt more confident interpreting output and tracking errors, especially when the system offered direct cues about causes of failure or success (P2, P10).
- Techno-overload was mitigated by the removal of excessive terminal noise and the addition of colour coordination and structured messaging, contributing to a cleaner and more manageable interface (P2, P10).

While layout changes introduced temporary disorientation for some users (P2, P13), it did not hinder performance, and familiarity returned with continued use (P10).

In terms of sentiment, participants expressed strong positive responses to the terminal improvements, particularly when they contributed to clearer feedback loops, quicker debugging, and reduced visual clutter. Minor usability concerns, such as initial confusion or preferences for terminal placement, did not significantly detract from overall effectiveness.

These findings reinforce that when terminal enhancements focus on visual clarity, execution feedback, and intuitive navigation directly support higher performance and minimise technostress. They also suggest that maintaining familiarity in layout and offering diagnostic detail in error cases can further empower users during runtime and debugging.

3.6.2.4 Breakdown by General

Among the clusters, based on participant feedback (n=5) the General cluster (Figure. 3.37) indicated notable improvement from the baseline (M = 79) to the modified version (M = 85.5). This suggests that the system modifications succeeded in refining the overall interaction quality, making the IDE more approachable and efficient for diverse user groups. The SEM decreased from 6.74 in Cycle 1 to 2.78 in Cycle 2, indicating improved precision in the estimated mean following the IDE modifications. However, the SEM ranges show slight overlap, suggesting that while usability scores increased, the improvement should be interpreted with caution and cannot be considered clearly statistically distinct based on SEM alone. Nevertheless, the reduced SEM in Cycle 2 indicates a more stable cohort response, reflecting a more consistent perception of usability improvements

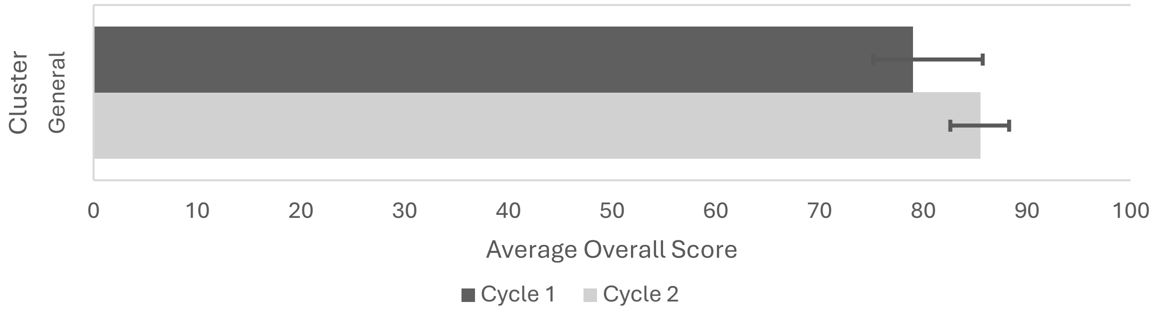


Figure 3.37: SUS Avg General Cluster Score

In the general cluster, where participants experienced a combination of modifications from all clusters (Figure. 3.38), performance and productivity achieved the highest scores (M = 5), followed by satisfaction (M = 4.6). For technostress triggers, techno overload saw the greatest reduction (M = 4.6), followed by techno-insecurity (M = 4.4). Reductions in techno-invasion and techno-complexity both scored (M = 4.2), with techno-invasion showing a smaller IQR, suggesting more consistent perceptions. Techno-uncertainty experienced the smallest reduction (M = 4). These results support the value of combining modifications from different clusters to enhance overall usability and reduce stress.

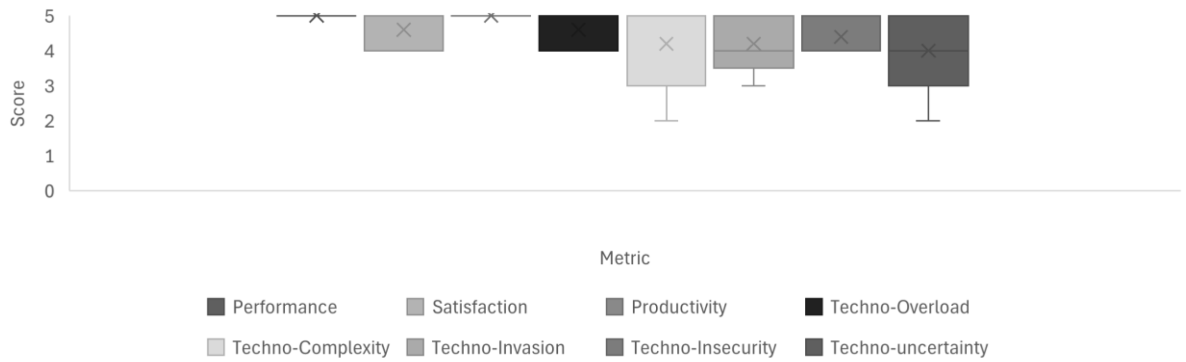


Figure 3.38: SMI for General Cluster

3.6.2.4.1 Lexical Salience

The lexical salience within the General cluster (Figure. 3.39) is dominated by code, feature, increased, and productivity, indicating that participants framed their overall experience around functional capability and efficiency gains. The prominence of increased is strongly associated with productivity and satisfaction, while references to techno-complexity and techno-overload suggest perceived reductions in cognitive strain. Terms such as assist, solution, help, and performance further indicate that supportive features and structured tooling were central to the broader positive evaluation of the system.

Figure 3.39: General Cluster Word Cloud

3.6.2.4.2 Sample Responses

3.6.2.4.2.1 Autocompletion

Participants strongly appreciated the inclusion of code assist tools such as AI-driven suggestions, autocomplete.

> *P1 noted that "to have the IDE have some level of understanding of the problem I'm facing, and be able to not only remind me of important syntax while I worked but suggest how I could use it in a way that contributes to the solution to my problem massively increased the speed I could work at".*
>
> *P11 emphasized the support provided at the beginning of the coding process "The tailored IDE provided a good starting point which I really appreciate… decreased my sense of techno-insecurity because I had a good starting point… the autocomplete are really impressive and captured the logic I needed almost perfectly which I appreciated".*
>
> *P9 shared that "My satisfaction has been increased since I find out better and more efficient ways of doing one thing thanks to some of the AI assistants… This modified IDE decreased techno-overload… techno-complexity… techno-insecurity, as I don't have to rely on my limited knowledge for the performance of a program, and could do as good as a veteran programmer if not better".*
>
> *P15 pointed to improvements for beginners, "Auto assist code increase the coding performance specially for beginners… Assistance bot help the productivity and save the time in trouble shouting".*
>
> *They further stated that it helped reduce multiple dimensions of technostress, "Auto assist code help and chatbot helps to decrease the techno-overload… new modified version also helped to reduce the techno-complexity… [and] sort-out this issue [of] techno-insecurity".*

3.6.2.4.2.2 Shortcut Menu

No notable comments were provided by this cluster regarding menu-related changes.

3.6.2.4.2.3 Error Handling

No notable comments were provided by this cluster regarding error handling-related changes.

3.6.2.5 Addressing Research Question Five

Question: How do tailored modifications to IDEs impact technostress – measured in terms of performance, productivity, and satisfaction? (RQ5)

Findings: The analysis revealed a positive impact of tailored IDE modifications on all three measured metrics, performance, productivity, and satisfaction across different interface areas. The distribution of improvements varied depending on the modified section of the IDE: (a) Satisfaction saw the greatest improvement in the menu, followed by the editor, and then the terminal, (b) Productivity increased the most in the editor, followed by the terminal, and lastly the menu (c) Performance also showed the highest gains in the editor, with the terminal and menu following. Interestingly, in the case of the terminal, both performance and productivity improvements were equally high, suggesting that modifications in this area had a particularly strong dual effect on how efficiently and effectively students worked.

Discussion: The findings support that tailored modifications to different areas of the IDE interface resulted in measurable, positive improvements across all three user experience metrics. This reinforces the value of a user-centred, feedback-informed approach to technostress-aware IDE design. Notably, the editor consistently ranked highest for improvements in both performance and productivity, while the menu played a more influential role in boosting satisfaction. These differences indicate that specific interface areas contribute differently to various dimensions of user experience, supporting the framework's strategy of customising modifications based on targeted feedback.

3.6.3 Internal Consistency

The internal consistency of the survey instruments was assessed using Cronbach's alpha across both experimental cycles. SUS in Cycle 1 indicated good internal consistency ($\alpha = 0.82$), indicating the reliability of participant responses. However, the SUS instrument in Cycle 2 produced a lower Cronbach's alpha value ($\alpha = 0.57$), indicating weaker internal consistency and suggesting that participant responses may have been less homogeneous within the modified IDE condition. This lower reliability may reflect increased variability in participant perceptions, differences in interpretation of usability-related questions, or the influence of individual preferences towards the modified IDE features. Consequently, the Cycle 2 SUS findings should be interpreted with caution and not considered in isolation.

To mitigate this limitation, SUS findings were triangulated with additional quantitative and qualitative measures, including TAM responses, SMI variables, behavioural observations, participant feedback, and biometric indicators. This broader multi-method evaluation approach helped provide additional contextual support when interpreting usability and user-experience outcomes beyond the SUS instrument alone.

The TAM survey showed excellent reliability across both cycles. In Cycle 1, TAM produced a Cronbach's alpha of 0.93, indicating high internal consistency. In Cycle 2, TAM again indicated strong reliability, with a Cronbach's alpha of 0.94. The SMI instrument, conducted only in Cycle 2, yielded a Cronbach's alpha of 0.78, indicating acceptable internal consistency among the variables, which included Performance, Satisfaction, Productivity, and the five Technostress triggers.

3.6.4 Task Assessment

When it came to understanding the task given all participants stated that the given task in both Cycle 1 (n=20) and 2 (n=20) made sense (100% Yes, 0% No). This highlights that the task design and instructions were effectively communicated, aligning with the participants' existing knowledge and skill levels.

In terms of difficulty (Figure. 3.40), for Task A, participants rated its difficulty higher (M = 2.7) in Cycle 1, which improved (M = 1.8) in Cycle 2. This significant reduction suggests that participants found Task A easier to complete when using the modified IDE. This reduction suggests that participants found Task A easier to complete when using the modified IDE. While this improvement may reflect the effectiveness of the enhanced IDE features in supporting participants to navigate and address task requirements more efficiently, it should be interpreted with caution as perceived difficulty may have been influenced by multiple factors. Although the study employed a crossover design to minimise learning effects through alternating task allocation between cycles, it is not possible to completely isolate the influence of the IDE modifications from familiarity gained through participation in the experimental process. Consequently, the reduction in perceived difficulty may represent a combination of IDE-related benefits, increased participant familiarity with the experimental environment, and task-specific characteristics. Conversely, Task B was rated at 2.2 in Cycle 1 and remained consistent (M = 2.2) in Cycle 2. This stability indicates that participants perceived Task B's difficulty as unchanged regardless of whether they used the baseline or modified IDE. The results suggest that the IDE modifications did not introduce additional complexity or hinder participants' ability to complete the task, indicating that the modified IDE was at least as effective as the baseline IDE in supporting Task B execution.

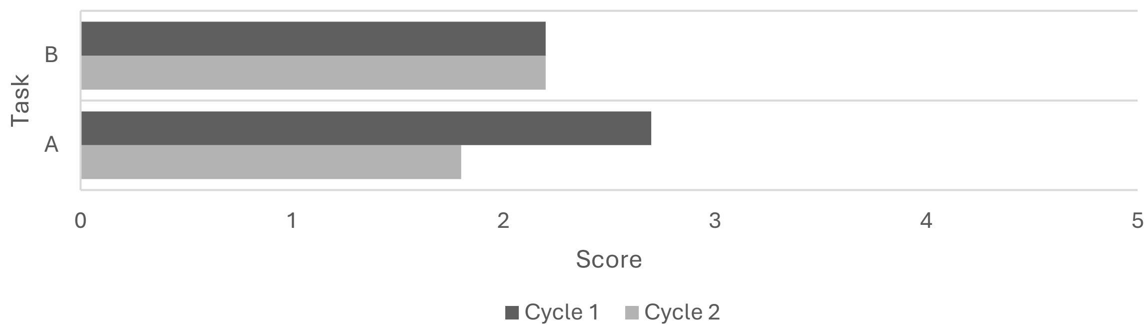


Figure 3.40: Individual Task Difficulty Comparison Across Cycles

When analysing the mean difficulty ratings across both cycles, the average score for Task A was slightly higher (M = 2.25) compared to Task B (Figure. 3.41), which had a marginally lower mean score (M = 2.2). These results indicate that, overall, both tasks were perceived as having relatively medium difficulty. The minimal difference in overall means also reinforces the idea that the tasks were well-balanced in terms of difficulty, achieving one of the experiment's key objectives. It further highlights that the modifications in the IDE are neither overly complicated nor drastically simplified task execution, ensuring a fair and consistent evaluation of the IDE's impact across different task types.

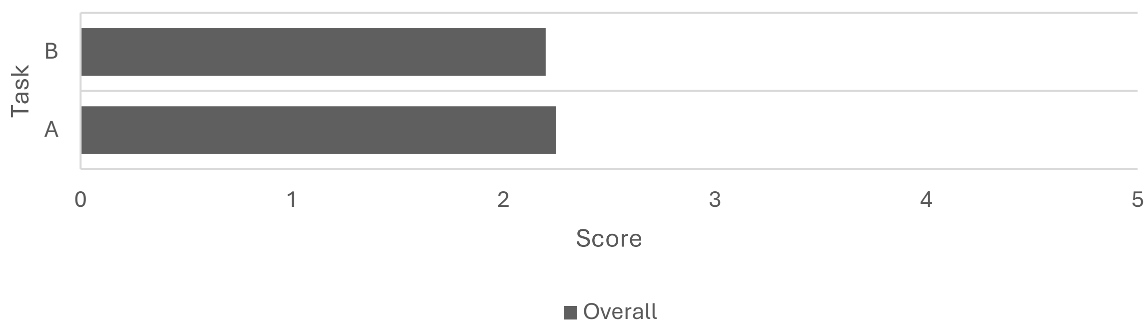


Figure 3.41: Task Difficulty Comparison Across Cycles

3.6.5 Task Outcomes

The TO findings (outlined in Table 3.19) highlight that the modifications introduced in Cycle 2 positively impacted productivity performance and satisfaction. The increase in task completion, along with higher test case success rates and improved speed, indicates an enhancement in efficiency, effectiveness and experience.

3.6.5.1.1 Productivity

Participant productivity improved from Cycle 1 to Cycle 2. In Cycle 1, none of the participants (n=20) completed the task within the allocated timeframe of 15 minutes. However, in Cycle 2, eight participants successfully completed the task, demonstrating an increase in task completion (TC).

3.6.5.1.2 Performance

Similarly, amongst those who completed the task in Cycle 2, the average completion time (TCT) was 14:08 minutes. In Cycle 1, the number of requirements met (TRQ) varied across participants, with some achieving as low as 0/7. By Cycle 2, several participants achieved 6/7 or 7/7, indicating improved task execution. In terms of test cases (TTC), most participants in Cycle 1 recorded 0/4 or 0/5 successful test cases. However, in Cycle 2, those who completed the task showed substantial improvements, with some achieving 4/4 or 5/5 successful test cases.

3.6.5.2 Satisfaction

All participants expressed a preference for the modified version (MV) from Cycle 2 of the IDE compared to the baseline from Cycle 1, indicating higher satisfaction with its features and usability. This preference outlines that the changes introduced in Cycle 2 contributed positively to the user experience, not only reducing technostress and improving workflow efficiency.

Table 3.19: Cycle 1 vs Cycle 2 Task Outcome (TO) Comparison

	Cycle 1					Cycle 2					
P	**TG**	**TC**	**TRQ**	**TCT**	**TTC**	**TG**	**TC**	**TRQ**	**TCT**	**TTC**	**MV**
1	A	N	0/7	N/A	0/4	B	Y	7/7	14:25	5/5	Y
2	B	N	6/7	N/A	5/5	A	Y	7/7	15:00	3/4	Y
3	B	N	3/7	N/A	0/5	A	N	6/7	N/A	0/4	Y
4	A	N	5/7	N/A	0/4	B	N	4/7	N/A	0/5	Y
5	B	N	5/7	N/A	0/5	A	Y	5/7	14:15	4/4	Y
6	B	N	6/7	N/A	0/5	A	Y	6/7	13:11	4/4	Y
7	A	N	6/7	N/A	0/4	B	N	6/7	N/A	0/5	Y
8	A	N	5/7	N/A	0/4	B	N	4/7	N/A	0/4	Y
9	B	N	1/7	N/A	0/5	A	Y	5/7	13:30	4/4	Y
10	A	N	1/7	N/A	0/4	B	N	4/7	N/A	0/5	Y
11	A	N	1/7	N/A	0/4	B	N	6/7	N/A	0/5	Y
12	B	N	4/7	N/A	0/5	A	N	6/7	N/A	0/4	Y
13	B	N	1/7	N/A	0/5	A	N	5/7	N/A	0/4	Y
14	B	N	4/7	N/A	3/4	A	Y	6/7	13:45	4/4	Y
15	A	N	2/7	N/A	0/5	B	N	6/7	N/A	0/5	Y
16	A	N	5/7	N/A	1/4	B	Y	6/7	15:00	5/5	Y
17	A	N	0/7	N/A	0/4	B	N	3/7	N/A	0/5	Y
18	B	N	2/7	N/A	0/5	A	N	2/7	N/A	0/4	Y
19	B	N	4/7	N/A	0/5	A	Y	5/7	14:05	4/4	Y
20	A	N	4/7	N/A	0/4	B	N	6/7	N/A	0/5	Y

Key: P - Participant, TG - Task Given, TC - Task Completion, TRQ - Task Requirements, TCT - Task Competition Time, TTC - Task Test Cases, MV - Modified Version

3.7 Discussion

This discussion is structured using a two-tier analytical hierarchy. At the cluster level (high-level analysis), quantitative findings are presented to show the aggregated impact of grouped IDE modifications (Menu, Editor, Terminal) on outcome variables and dominant technostress dimensions. This level reflects the measurable, statistically observable effects across each intervention cluster.

At the feature level (low-level analysis), qualitative findings are examined to identify how individual modifications influenced specific techno-stressors (Table. 3.20). Each feature may reduce or temporarily elevate distinct dimensions such as Techno-Overload, Techno-Complexity, Techno-Insecurity, Techno-Uncertainty, or Techno-Invasion.

Crucially, reductions observed at the feature level do not remain isolated. When multiple features within a cluster collectively reduce related techno-stressors, their combined effect may manifest at the cluster level as a dominant reduction in a broader technostress dimension. Thus, low-level technostress adjustments aggregate into high-level technostress outcomes.

This hierarchical interpretation ensures that quantitative cluster-level findings represent the cumulative manifestation of feature-level technostress dynamics, thereby preserving analytical coherence between micro-mechanisms and macro-outcomes.

Table 3.20: Mapping Table

Area	Modification	Pe	Pr	Sa	T-Ov	T-Co	T-Iv	T-In	T-Un
Menu	Shortcut			✓	↓			↑	
	Clutter			✓	↓			↑	
	Packages		✓			↓		↑	
Editor	Code Assist	✓			↑				↓
	Syntax			✓		↑	↓		
	Snippets		✓		↓			↑	
	Themes			✓	↓			↑	
Terminal	Runtime	✓			↑			↓	
	Errors			✓		↑			↓
	Layout			✓			↑		↓
	Output	✓			↓	↑			

Key: ✓ – Predominantly Affected, ↓ - Predominantly Decreased Trigger, ↑ - Predominantly Increased Trigger, Pe – performance, Pr – Productivity, Sa – Satisfaction, T-Ov – Techno-overload, T-Co- Techno-complexity, T-Iv – Techno-Invasion, T-In – Techno-insecurity, T-Un – Techno-Uncertainty

3.7.1 Menu Mapping

3.7.1.1 Cluster Level

Mapping: Menu → Satisfaction → Techno-Overload, Techno-Insecurity

Menu-related changes, including shortcut enhancements, clutter reduction, and the addition of a package manager, predominantly contributed towards improved satisfaction by making navigation more intuitive and visually clean. The predominant reduction observed was in Techno-Overload, as simplifying access points and reducing noise lowered cognitive strain and improved focus. A notable reduction was also observed in Techno-Insecurity, as clearer structure and improved accessibility increased users' confidence in navigating and managing the IDE environment. Simplifying access points and reducing noise allowed users to feel more in control without being overwhelmed by interface complexity.

3.7.1.2 Feature Level

The shortcut menu feature primarily enhanced performance by enabling quicker access to commonly used functions, which boosted efficiency. It predominantly reduced techno-overload by simplifying the navigation process and reducing the cognitive load of searching for tools. However, it could potentially increase techno-insecurity for users who are unfamiliar with the new layout and struggle to adapt to the changes.

Clutter reduction notably improved performance by streamlining the interface and making the IDE more focused and task oriented. This reduction in visual clutter predominantly decreased techno-overload, as fewer distractions allowed users to focus better on their tasks. However, it could increase techno-insecurity for users who are accustomed to a more feature-rich environment and may feel uncertain about the lack of options available.

The package manager feature contributed to productivity by simplifying the management of dependencies, allowing users to focus on coding rather than troubleshooting installations. This change predominantly reduced techno-complexity by making package management more accessible and visually organised, enhancing the overall user experience. However, it could potentially increase techno-insecurity as some users might still feel unsure about their full understanding of package management processes, especially if they rely heavily on the visual aids provided by the IDE.

3.7.2 Editor Mapping

3.7.2.1 Cluster Level

Mapping: Editor → Productivity → Techno-Invasion

Editor features like autocompletion, code snippets, syntax highlighting, and theme customisation predominantly elevated productivity by supporting efficient code creation and personalisation. These enhancements led to a notable reduction in Techno-Invasion, as users experienced fewer intrusive disruptions and less workflow interruption during coding tasks. Tools that scaffolded code writing and improved visual clarity made the environment feel more seamless and less interruptive, allowing users to maintain task focus and continuity, especially for those with moderate experience.

3.7.2.2 Feature Level

Autocompletion primarily enhanced performance by speeding up coding tasks, while it reduced techno-uncertainty by offering contextual suggestions, but can also predominantly increased techno-overload due to intrusive or incorrect completions that disrupted workflow.

Syntax highlighting primarily enhanced satisfaction by improving visual clarity and ease of use, while it reduced techno-invasion by minimising error-related stress and supporting quicker code comprehension; however, it could potentially increase techno-complexity for users unfamiliar with the meaning behind varied colour schemes.

The use of templates and code snippets predominantly improved productivity by providing structural support and reducing initial coding effort, while they predominantly reduced techno-overload by eliminating the need to start from scratch; however, they could also increase techno-insecurity for users who may feel dependent on them or uncertain when such support is absent.

The theme customisation features primarily enhanced satisfaction by improving emotional comfort and providing a more personalised experience. It also reduced techno-overload by allowing users to tailor the IDE to their preferences, thus reducing cognitive strain. However, it could increase techno-insecurity as users might compare their themes with others, leading to feelings of inadequacy.

3.7.3 Terminal Mapping

3.7.3.1 Cluster Level

Mapping: Terminal → Performance → Techno-Complexity, Techno-Uncertainty

Terminal modifications including runtime indicators, structured error handling, alternative layouts, and minimalist output predominantly boosted performance by improving code execution quality and speed. The most significant technostress reduction was in Techno-Complexity, as these features demystified the runtime process and debugging, making technical operations easier to follow. A notable reduction was also observed in Techno-Uncertainty, as clearer feedback and improved execution visibility increased user confidence in understanding system behaviour. Cleaner terminal outputs and feedback loops simplified what is traditionally seen as a dense, code-heavy interface area.

3.7.3.2 Feature Level

Runtime indicator notably improved performance by providing clear, concise feedback on program execution, enhancing task fluency and reducing confusion. This visibility reduced techno-insecurity, as users felt more confident and informed about the program's status. However, it could increase techno-overload for some users due to the added details such as exit codes and runtime metrics, which may overwhelm those preferring simpler outputs.

Error handling notably improved satisfaction by delivering clear and actionable feedback, making debugging more intuitive and reducing the time spent on identifying issues. This structure lowered techno-uncertainty, as users felt more confident in understanding and resolving errors. However, it could increase techno-complexity for users who may find detailed terminal outputs dense or intimidating, especially if unfamiliar with interpreting such messages.

Terminal layout had a mixed impact on satisfaction, with some users appreciating the eventual usability, while others expressed initial discomfort due to unfamiliar placement. Despite these differences, the feature did not significantly hinder task performance. It helped reduce techno-uncertainty over time as users acclimated to the new layout and suggested improvements for smoother interaction. However, it could increase techno-invasion initially, especially for users disrupted by non-traditional configurations or unexpected focus shifts during execution.

Minimalistic output significantly improved performance by decluttering terminal results and making outputs easier to interpret, which enhanced task focus and emotional ease. This streamlined presentation reduced techno-overload by eliminating unnecessary information and improving visual clarity. However, it could increase techno-complexity for new users who might feel confused without proper onboarding or explanation of the simplified output format.

3.7.3.3 Addressing Research Question Six

Question: Which features of the modified IDE contribute to reducing technostress factors? (RQ 6)

Findings: The mapping analysis (Table 3.20) indicates that menu-related modifications (shortcut restructuring, clutter reduction, and package management integration) exerted the strongest impact on reducing techno-overload and techno-insecurity, primarily by lowering cognitive load and improving navigational clarity. Editor-level enhancements (e.g., autocompletion, snippets, syntax highlighting, and customisation) were most influential in reducing techno-invasion, as they minimised disruptive friction during task execution and supported uninterrupted workflow. Features targeting structural simplification and dependency management contributed notably to reductions in techno-complexity, while uncertainty-related reductions were more modest and typically secondary to improvements in clarity and control. Terminal-related modifications, particularly improved command visibility, structured output formatting, and integrated feedback mechanisms, contributed most notably to reductions in techno-complexity and techno-uncertainty, as they clarified system responses, improved transparency of execution processes, and reduced ambiguity during debugging and task execution.

Discussion: While the mapping identifies the primary technostress trigger each modification is intended to address, it is important to note that, due to the interconnected nature of these triggers, a single modification can often contribute to the reduction of multiple forms of technostress. For example, a feature designed to reduce complexity might also alleviate feelings of overload or uncertainty. However, to maintain analytical clarity, this study emphasises the most prominent associations supported by experimental data. These predominant links were identified through a combination of participant feedback, usage data, and performance outcomes, offering a focused yet robust view of how specific IDE enhancements can alleviate key sources of technostress in student developers.

3.8 Threats to Validity

3.8.1 Internal

IT1. ***Subjective Interpretation of Survey Constructs***: Unlike SUS and TAM, which employ well-established and tightly defined items, both the UEI and SMI include constructs that may be interpreted subjectively by participants depending on their experience level and familiarity with technostress-related terminology. Variations in interpretation may therefore have influenced participant responses and introduced measurement variability.

IT2. ***Open-Ended Response Interpretation***: Open-ended questions were included to capture richer participant perspectives. However, qualitative responses are inherently influenced by question wording and individual interpretation. Differences in how participants understood or articulated concepts such as stress, difficulty, or system impact may have introduced variability unrelated to actual IDE-induced effects.

IT3. ***Learning Effects***: Although a crossover experimental design was employed to reduce ordering effects and balance task allocation, learning effects cannot be completely excluded. Participants may have become more familiar with the experimental environment, programming workflow, or assessment process between cycles, potentially influencing performance, perceived workload, and task-difficulty ratings. Consequently, reductions in perceived difficulty observed during Cycle 2 may not be attributable solely to the IDE modifications, but may instead reflect a combination of IDE enhancements, participant familiarisation, and task-specific characteristics.

IT4. ***Order Effects***: The study employed a within-subjects design in which all participants completed Cycle 1 using the baseline IDE before progressing to Cycle 2 using the modified IDE. This approach was selected to allow direct comparison of participant responses across both IDE versions while controlling for individual differences in programming ability, experience, and emotional disposition. To reduce potential practice effects, two different programming tasks were used and alternated between participant groups across cycles. However, although this design helped minimise learning effects associated with repeated exposure to the same task, it could not eliminate them entirely. Familiarity with the experimental procedure, development environment, or assessment process may have influenced participant performance and perceptions during Cycle 2. Consequently, the absence of full counterbalancing represents a limitation of the study, and some observed improvements may partially reflect order effects rather than IDE modifications alone.

IT5. ***Statistical Robustness***: Formal assessment of distributional assumptions was not undertaken, and therefore the extent to which the data satisfied assumptions commonly associated with statistical analyses cannot be fully verified. As a result, findings should be interpreted alongside the qualitative evidence and viewed as exploratory rather than definitive. Future studies should employ larger samples and conduct formal assumption testing where appropriate.

IT6. ***Task Comparability***: Although Task A and Task B were designed to be comparable in terms of programming requirements, expected completion time, implementation complexity, and associated test cases, no formal validation study was conducted to establish task equivalence. Consequently, differences in participant perceptions of task difficulty may have influenced performance, usability, emotional, or technostress-related measures. While task alternation was used to reduce this threat, it is possible that some improvements observed in Cycle 2 were influenced by task-specific characteristics and/or order effects rather than IDE modifications alone.

IT7. ***Measurement Instrument Selection***: While the study incorporated established instruments including SUS and TAM, alongside the custom SMI instrument, it did not employ a dedicated technostress assessment instrument such as the Technostress Creators Inventory [306]. The selected instruments were chosen to evaluate usability, technology acceptance, perceived performance, productivity, satisfaction, and technostress-related factors directly associated with IDE interaction and modification impact. However, the absence of a validated technostress-specific instrument may limit direct comparability with portions of the broader technostress literature. Consequently, findings relating to technostress should be interpreted in conjunction with the usability, acceptance, qualitative, and biometric evidence collected throughout the study. Future research should consider incorporating validated technostress instruments alongside usability, technology acceptance, and task-outcome measures to provide a more comprehensive assessment of technostress within programming environments.

3.8.2 External

ET1. ***Demographic Representation***: While gender, education level, and programming experience were examined, other potentially relevant characteristics, such as age, disability status, and nationality, were not incorporated into comparative analyses. This decision was primarily due to the uneven or limited distribution of these characteristics within the sample, which constrained meaningful statistical comparison. Future research should aim to recruit more demographically diverse and balanced samples to enable a more comprehensive examination of how these factors may moderate technostress responses and IDE interaction patterns.

3.9 Summary

This chapter addresses a critical gap in existing research, where user performance, productivity, and satisfaction are negatively impacted by the lack of solutions aimed at mitigating technostress in IDEs. In response, this study contributes a novel framework for technostress-centric IDEs, leading to the development of Technostress-Integrated Development Environments (TIDEs). The proposed framework supports its potential by achieving two key objectives: reducing technostress among developers (Obj 1) and enhancing core user outcomes such as performance, productivity, and satisfaction (Obj 2).

In terms of gender, male participants responded more positively to the tailored IDE compared to females. While female users generally prioritise usability and ease of use, male users tend to value performance and functionality, resulting in more consistent feedback when these needs are met (RQ1). Educational level also influenced the perception of IDE modifications, with the most notable benefits observed among students in the earlier stages of their undergraduate studies (RQ2). This trend was similarly reflected in participants with less programming experience, suggesting that tailored enhancements were particularly effective in supporting novice users (RQ3). Ethnic background further shaped user experience, with Black participants reporting the most substantial improvements in usability and acceptance following the modifications (RQ4).

Although improvements in usability and technology acceptance were observed, these findings should not be interpreted as definitive evidence of technostress reduction. Rather, they provide supporting evidence that complements the technostress-related findings obtained through the SMI instrument and qualitative feedback. Consequently, any conclusions regarding technostress are based on the convergence of multiple sources of evidence rather than usability outcomes alone.

The SMI analysis suggests that tailored IDE modifications positively influenced all three measured metrics, performance, productivity, and satisfaction across the various interface areas, while also contributing to a reduction in all five dimensions of technostress (RQ5). Finally, the study's mapping of IDE modifications to specific outcomes and technostress triggers (RQ6) offers practical insights for designing more technostress-aware development environments. These mappings highlight how targeted enhancements can simultaneously boost performance and productivity while also fostering greater user satisfaction and reducing various forms of technostress.

Building on this foundation, Chapter 4 focuses on the construction and structure of the TIDE dataset, which consolidates the biometric, and behavioural, data generated during the study, explaining how the dataset is designed to support subsequent analysis of technostress, emotional response, and IDE interaction patterns.

Chapter *Four*

4 TIDE-Dataset

4.1 Introduction

This chapter introduces TIDE dataset, a novel, multimodal dataset developed to capture and analyse emotional and cognitive responses during programming tasks. The dataset builds directly upon the Iterative IDE Enhancement Framework introduced in the previous chapter, complementing and serving as its empirical foundation by providing the quantitative and biometric evidence needed to evaluate how emotions fluctuate during real IDE use.

The TIDE dataset represents one of the first structured attempts to combine facial emotion recognition and user interaction data with contextual task information in programming environments. It captures real-time biometric signals that reflect users' emotional states while simultaneously recording performance-related data across key programming phases (Analysis, Implementation, and Testing). By aligning these physiological and behavioural markers with qualitative feedback and survey responses, TIDE enables a deeper, evidence-based understanding of how users experience emotions and how IDE design decisions influence technostress and outcomes.

This chapter outlines the design, structure, and composition of the dataset, describing the data collection methodology, the experimental setup, and the breakdown of the emotional markers across cycles and phases. It also details the link between biometric features and derived metrics (performance, productivity, and satisfaction), which were identified in Chapter Two as key outcome indicators for assessing the impact of software design decisions on user wellbeing. To complement the framework introduced in Chapter Three, this dataset provides an empirical baseline for comparison against the data collected during framework evaluation, ensuring consistency between the conceptual, applied, and data-driven components of the research.

Participant pathways underpinning this study (illustrated in Figure. 4.1), link the experimental procedures in Chapter Three, and the biometric analysis in this Chapter. The participant pool of undergraduate computing students completed the baseline and modified IDE tasks with post hoc surveys and biometric capture.

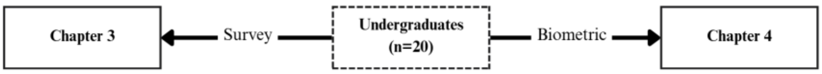


Figure 4.1: Participant Pathway for Post-Hoc Surveys and Biometric Capture

Structuring the evaluation across two chapters is intentional and methodologically grounded. Chapter Three presents the post-hoc survey findings first because these reflections capture participants' conscious, self-reported perceptions after completing each IDE cycle. These data provide an essential subjective baseline for understanding how participants interpreted their own experience.

This Chapter then presents the intra-experiment biometric data, which measures participants' affective responses as they occurred during the task rather than through later reflection. Separating these chapters ensures conceptual and analytical clarity: Chapter Three addresses what participants believed or felt they experienced, while This Chapter captures what their behavioural and emotional signals objectively revealed. Presenting the survey results first also allows the subsequent biometric analysis to be interpreted in relation to user-reported perceptions, highlighting convergence, divergence, and deeper behavioural patterns that may not be accessible through surveys alone.

4.1.1 Problem Statements and Research Questions

The aim of this chapter is to explore facially expressed emotional responses recorded during IDE-based programming tasks undertaken by computing higher education students, examining how these emotions evolve across task execution phases and how they relate to task outcomes. The dataset supports structured analysis in support of the following objectives:

- (Obj 1) To provide a reusable emotional dataset that supports research into technostress-centric user interface design.
- (Obj 2) To identify representative emotional responses by analysing how specific emotions correlate with and encapsulate patterns observed in other emotional states during programming tasks.

To achieve these objectives, the study addresses the following research questions:

- (RQ 1) What datasets aid in decision making regarding technostress?
- (RQ 2) Which emotions are representative of others with regards to expressing technostress?

RQ 1 (Obj 1) examines how the resulting dataset can be utilised as an evidence base for guiding technostress-related design decisions and future technostress-aware system development. RQ 2 (Obj 2) examines the relationships and patterns among emotional responses observed during programming activity, providing insight into how emotions interact and evolve throughout task execution.

4.1.2 Contribution and Significance

The TIDE dataset contributes a unique new resource to the fields of software engineering, HCI, and affective computing. The research is centred around biometric data captured during real-world coding tasks, with emotional indicators grounded in widely recognised psychological theories.

The key contributions aligned with the research questions are as follows:

- (C1) A Novel Dataset (RQ1): Presenting a first-of-its-kind (to our knowledge) time-series dataset that captures facial emotional and gaze states across distinct phases of a programming task within an IDE. The dataset includes rich time-based labelling, phase segmentation, and demographic metadata, making it suitable for a wide range of HCI and affective computing studies.
- (C2) Representative Emotional Markers (RQ2): Utilising a replicable method to identify representative emotional markers, offering theoretical and experimental grounding for emotion-centric dynamics in development workflows. Using standardised tools (Correlation Matrices, PCA, and MANOVA), to derive the emotional markers and laying the groundwork for repeatable emotion-based evaluations in similar software settings.

The novelties of this contribution are as follows:

- (N1) Introduction of biometric technostress data into software design: TIDE dataset offers a concrete resource for bridging emotional response data with practical interface outcomes in IDE use. By linking user experience and emotion in development contexts, the dataset supports nuanced exploration of how emotional engagement and regulation change across phases and cycles and affect user performance, productivity and satisfaction as well as user task outcomes, offering insights into designing for technostress-centric user interface design.

4.1.3 Structure

Section 4.3 reviews prior research on the use of biometric data and highlights the relatively little work that has focused on capturing and interpreting emotional responses in the domain of interest. This section also introduces established emotion theories as useful tools for interpreting facial-emotion markers. Section 4.4 outlines the end-to-end process, including the tools and setup used for data collection. Sub-section 4.4.2 presents the data analysis process and introduces the TIDE dataset, a novel biometric dataset that captures facial-emotion data during software development tasks. It explains the structure of the dataset, its sub-datasets, and illustrates the potential applications in affective computing, technostress research, and emotional analysis in digital tool usage. Section 4.5 describes the experimental procedure adopted throughout the cyclical process. Section 4.6 discusses the key findings, including how emotional expressions varied across development phases and task cycles. It also explores correlations between emotional markers and changes in emotional patterns that influence task outcomes. Section 4.7 presents the major insights by addressing each of the two research questions and outlining the study's limitations. It distinguishes between internal and external threats to validity and considers the extent to which the findings may generalise beyond IDE-focused contexts. Section 4.8 concludes major insights in relation to the chapter's objectives. Finally, Section 4.9 outlines short- and long-term potential on building up on this foundational research.

4.2 Background

Mental health can be negatively impacted by technostress, which results from the constant use of technology [307]. Given its negative connotations, if not addressed, technostress can have adverse physiological effects such as increased stress, affecting the brain, emotions, and body which can also have knock on effect on satisfaction and health might suffer with technology use causing unnecessary stress [308], [309]. Promoting digital well-being and providing emotional provisions can be a way to help cope with digital stress.

4.2.1 Emotions

Many factors influence what emotions are and how they are perceived. As a result, asking people about their feelings can be challenging because verbal reports appear to be influenced by one's knowledge of inner states, cultural impact, and linguistic skill [310]. One way to address this limitation is through the use of physiological measures, which provide a more universal and objective means of assessing emotional responses compared to self-reported data [311]. Bodily reactions can be measured objectively by tracking, Brain Activity (EEG), Heart Rate (ECG), Electromyography (EMG), Skin Conductance (GSR) and Facial Expressions (FE). Discussed below are the quantitative and qualitive approaches which will be utilised to capture emotional and behavioural response from participants.

Although the terms emotion and feeling are often used interchangeably, they denote distinct but interdependent constructs. Emotions are automatic physiological and neurobiological responses to internal or external stimuli, often occurring outside conscious awareness [312]. They involve bodily and neural mechanisms that prepare individuals for adaptive action. Feelings, on the other hand, represent the conscious, subjective interpretation of those emotional states, the internal awareness that emerges when one reflects upon an emotion [313]. Thus, while emotions can occur without conscious recognition, feelings cannot exist independently of emotions [314]. This distinction between unconscious emotional processes and consciously experienced feelings provides a crucial foundation for understanding affective states in both psychological and computational research.

Emotions are widely recognised as pre-cognitive processes that occur prior to deliberate reasoning or conscious thought [315]. This rapid response system has a critical evolutionary advantage, in situations involving potential threat or urgency, emotions trigger immediate physiological and behavioural reactions, allowing individuals to act in split seconds without the need for reflective cognition [316]. Such mechanisms, often mediated by the amygdala, prioritise survival by bypassing slower cortical processing routes. Beyond their adaptive role in survival, emotions also shape cognitive functions, influencing attention, learning, and decision-making [317]. As [318] suggests, emotion is not the opposite of reason but rather a fundamental component of it, serving as both a motivational and evaluative force that guides behaviour and goal-directed thought.

4.2.2 Exclusions

EEG monitoring was not included in this study for several methodological and ethical reasons. Firstly, EEG requires the physical placement of electrodes on a participant's head, which can cause discomfort or distraction, potentially influencing emotional or cognitive responses during the experiment. Moreover, given the diversity of participants involved, factors such as cultural norms, personal preferences, and individual sensitivities could affect their willingness to participate under such conditions, introducing inconsistency and potential bias. To maintain a uniform experience and ensure participant comfort across all sessions, EEG monitoring was therefore excluded. While EEG can yield valuable insights into neural activity and cognitive states, its exclusion in this context ensured consistency, fairness, and reliability across all participants, while still allowing for meaningful interpretation through non-invasive biometric measures.

Similarly, GSR and EMG measurements were also excluded due to practical and technical constraints. Participants were required to perform continuous keyboard-based programming tasks, which can interfere with the accurate detection of skin conductance and muscle activity. Movement artefacts from typing and hand positioning often distort EMG readings, while GSR sensors are highly sensitive to environmental factors such as light, temperature, and motion. These limitations led to unstable and unreliable data during pilot testing. Consequently, to preserve data validity and minimise noise, GSR and EMG measurements were omitted. Although both sensors can provide valuable insight into physiological arousal and muscular tension, their exclusion prioritised methodological consistency and ensured that the study's biometric analysis based on facial emotion tracking remained robust, reliable, and non-intrusive.

4.2.3 Facial Expression Analysis

Emotions are central to human experience, influencing perception, decision-making, memory, and interpersonal interaction [319]. Among all the non-verbal cues humans display, the face serves as one of the most powerful indicators of emotional state. Subtle changes in key facial features, such as the eyes, eyebrows, mouth, and nostrils, communicate affective information that can be automatically interpreted by others or, increasingly, by computational systems [320]. FE therefore act as a dynamic and accessible window into a person's internal emotional state, offering measurable insights into both positive and negative affective responses.

Foundational work in affective computing has established facial expressions as a non-intrusive modality for capturing spontaneous affective responses during HCI [5]. Building on this foundation, Facial Expression Analysis (FEA) is employed in this research as a real-time, non-invasive approach for quantifying emotional responses during programming tasks performed within IDEs. By capturing FE data during active task engagement, it becomes possible to identify fluctuations in effect, that may correspond to specific IDE interactions or cognitive challenges. This provides an additional, objective layer of emotional insight beyond self-reported data.

Biometric sensing, and FEA in particular, has been extensively explored within the broader field of affective computing to infer users' emotional and cognitive states during interaction (such as deception detection and criminal interrogation to capture real human emotional responses directly from video data rather than self-report). Examples include, FacialCueNet model which was developed by [321] to detect deception by analysing facial cues including action-unit frequency, symmetry, gaze patterns, and micro-expressions extracted from recorded videos, achieving meaningful performance in distinguishing deceptive from truthful responses using a FE dataset collected under conditions similar to actual criminal interrogations A pilot study by [322] explored whether FEA with automated tools such as FaceReader can reveal emotional responses during polygraph examination questions, showing the potential of FEA for detecting emotional and cognitive states linked to deception when compared with classical physiological measures. A similar study by [323] analysed fear-related facial action units (extracted via computer vision) from video clips of participants involved in deceptive or truthful responses, showing that specific FE features differ between conditions.

While FEA has been extensively studied across various domains its primary focus tends to be on the accurate classification of emotional states independent of task context [324] [325]. More recent studies have suggested that FE can serve as indicators of cognitive and affective processes during task execution [7]. [326] conducted an experiment in which where participants interacted with typical computer tasks while their facial expressions were recorded and analysed to infer affective states, demonstrating the use of FEA in active software interaction contexts rather than passive stimuli alone. Similarly, [327] investigated adaptive user interface design by capturing real-time facial expressions to recognise emotions and adjust the interface accordingly in a user study. [328] analysed students' FE continuously during classroom interaction to assess emotional dynamics such as happiness, fear, and surprise using FEA, illustrating how biometric emotion capture informs understanding of engagement and affective experience in an educational context.

These works indicate that biometric FEA has been deployed in experiments with real participants to support inference of internal states such as deception, confidence, or emotional response. They provide robust precedents for participant-based biometric data collection, showing that FEA can reveal meaningful behavioural cues in contexts beyond surveys and background datasets. While informative, such approaches typically do not account for the fine-grained, phase-dependent nature of cognitively demanding activities such as programming, nor do they explicitly link emotional responses to tool usage, task difficulty, or technostress-related factors.

4.3 Related Work

This section reviews prior research on the use of biometric data and highlights the relatively little work that has focused on capturing and interpreting emotional responses in the domain of interest (thereby setting the context for Obj 2). It also presents established emotion theories as useful tools for interpreting facial-emotion markers, which sets the context for the novel dataset (addressing Obj 1).

4.3.1 Programming Domain

While biometric data has increasingly been adopted in software engineering and HCI research, existing studies largely focus on cognitive and behavioural dimensions, rather than the emotional experience of programming. A study by [329] conducted a combined EEG and ET experiment to investigate programmer efficacy (gaze patterns and neural correlates of cognitive load) during program comprehension tasks. This study explicitly links biometric measures to programming behaviour and performance. [330] used EEG to assess cognitive load associated with code comprehension, exploring changes in brain activity linked to code difficulty. The study also argues for combining EEG with ET for deeper insights, showing multi-modal biometric approaches in programming research. [331] investigated task difficulty classification in software testing by combining EEG, ET, and electrodermal activity (EDA) across multiple sensors. Similarly, [332] used biometric data to study how developers understand code, employing eye-tracking to analyse visual attention during comprehension tasks. The emphasis here was on cognitive processing and navigation through code rather than how the developer felt during the task. Likewise, [333] also applied ET to investigate novice programmers' visual attention during debugging and code writing. These studies offer insights into attention and comprehension patterns but do not address the emotional states that may underlie those behaviours. Similarly [334] investigated how developers navigate realistic code change tasks using ET to analyse attention patterns. Their findings showed that developers concentrated on specific parts of a method that were often connected through data flow. These studies, while valuable, treat biometric data as a proxy for performance or focus, not as a window into emotional engagement or technostress. Overall, none of these studies provide a reusable, phase-segmented dataset combining biometrics and detailed task outcomes, which is precisely the role the dataset in this chapter aims to fill. The TIDE dataset differs from prior FE research by embedding FE analysis directly within IDE-based programming tasks and aligning emotional markers with task phases, interaction events, and outcome metrics. Rather than treating facial expressions as standalone indicators of affect, TIDE captures spontaneous emotional responses as they emerge during authentic programming activity and uses them to inform technostress-aware interface evaluation.

4.3.1.1.1 Summary

A survey conducted by [335] on classification of technologies for measuring the cognitive load of developers highlighted how there was numerous studies using EEG, ET or a combination of mixed sensors but not any found for FEA.

Overall, although aligning with the programming domain of this study (as Table 4.1 presents), none of these studies provide a reusable, phase-segmented dataset combining biometrics and detailed task outcomes, which is precisely the role the dataset in this paper aims to fill. The TIDE dataset differs from prior research by embedding FEA directly within IDE-based programming tasks and aligning emotional markers with task phases, interaction events, and outcome metrics. Rather than treating FE as standalone indicators of affect, TIDE captures spontaneous emotional responses as they emerge during authentic programming activity and uses them to inform technostress-aware interface evaluation.

Table 4.1: Existing Related Work vs Presented Research

Study	Data Type	Phases	DM	TO	SS	TS
Existing Work						
[329]	EEG & ET		**			**
[330]	EEG & ET		**			**
[331]	EEG, ET & EDA					**
[332]	ET					**
[333]	ET		*			**
[334]	ET		*			**
This Paper						
- TIDE	FEA & ET	**	**	**	**	**

*Key: ** - yes, * - partial (e.g. only programming experience), DM – Capturing of Demographic Data, TO – Capturing of Task Outcomes, Capturing of Statistical Summary, TS – Time Series Data Captured*

4.3.2 Emotional Frameworks

To understand user affect during software interaction tasks, especially in cognitively demanding environments such as IDEs, it is essential to ground facially expressed emotional data in robust psychological theory. This study draws upon five complementary emotional frameworks to interpret facial emotion patterns across software use phases and design contexts. Together, these models offer a comprehensive foundation for interpreting emotion as an interplay of expression, cognitive interpretation, motivational state, and functional adaptation, enabling a multi-layered understanding of the emotional landscape within software-based task performance.

4.3.2.1 Basic Emotions

William James' early work in 1890 proposed a foundational set of basic emotions [264]. Building on this, Richard (1996) expanded emotional categorisation by suggesting a broader set of fifteen distinct emotions [265]. Later, Plutchik's Wheel of Emotions, introduced in 2001 [266], conceptualised emotions as eight primary categories with twenty-four finer-grained emotional states, a model that also encompasses the core emotions identified by Ekman. Paul Ekman's influential theory, originally proposed in 1992 [336] defines six universal emotions; anger, fear, joy, sadness, surprise, and disgust, which are considered evolutionarily ingrained and consistently expressed across cultures. These emotions are among the most reliably detected through automated facial recognition systems and serve as the empirical basis for emotion classification in this study. Ekman's framework is particularly valuable for moment-to-moment emotional tracking during interactive tasks. While it does not address more cognitively complex emotions like confusion or engagement, it provides the necessary expressive grounding for real-time affective measurement in dynamic environments. Accordingly, within this study, Ekman-based emotion categories should be interpreted primarily as practical classification labels for observable facial-expression markers rather than absolute indicators of participants' internal psychological experiences. This distinction is particularly important within programming and problem-solving environments, were cognitive states such as concentration, uncertainty, mental workload, or analytical reasoning may overlap with conventional emotional classifications. Consequently, the emotional outputs generated within the TIDE dataset are intended to support contextual behavioural analysis and affective trend identification rather than definitive psychological diagnosis or emotion verification.

4.3.2.2 Appraisal Theory

Richard Lazarus [265] emphasised the cognitive appraisal of situations as the basis for emotional responses. Emotions, in this model, arise when individuals evaluate an event's relevance to their goals, their capacity to cope, and the potential outcomes. As conceptualised by Lazarus and Folkman [337] "*Cognitive appraisal can be most readily understood as the process of categorizing an encounter, and its various facets, with respect to its significance for well-being.*" In delineating this categorization process, two major classes of appraisal: primary appraisal, which is an evaluation of what is at stake in the encounter and secondary appraisal, which is an evaluation of options and resources for coping with a stressful encounter. In describing coping [338] have differentiated between two basic functions, corresponding to two different types of coping: Problem-focused coping refers to "*the management or alteration of the person–environment relationship that is the source of stress*" whereas emotion-focused coping refers to "*the regulation of stressful emotions*" that arise in response to the problem. In software development tasks, Lazarus' theory can help explain emotional shifts as goal-relevant appraisals of task difficulty, coping ability, and interface responsiveness framing affect as a dynamic response to perceived control and challenge.

4.3.2.3 Control-Value Theory

Pekrun Control-Value Theory (CVT) [339] focuses on emotions arising within achievement contexts, where affective responses are shaped by an individual's perceived control over success or failure and the subjective value they assign to the task. Positive emotions promote engagement, while negative emotions can hinder it [340].. Some studies suggested that negative emotions promote the usage of metacognitive strategies and positively predict learning performance [341], [342], this phenomenon indicated that negative emotions may not reduce satisfaction. Emotions that emerge either during achievement-related processes (e.g., learning or problem-solving) or in response to achievement outcomes (e.g., test performance or task completion) [343], [344] are defined as achievement emotions [345]. This theory is particularly relevant for interpreting how users emotionally respond to coding tasks in tailor-based IDE settings. It supports a deeper understanding of how perceived progress contribute to emotion regulation and performance.

4.3.2.4 Circumplex Model of Affect

James Russell Circumplex Model [346] proposed that emotions can be situated within a two-dimensional space defined by valence (positive to negative; x-axis) and arousal (low to high; y-axis) [347]. This model supports dimensional clustering, enabling the grouping of related emotions into broader affective categories and identifying transitions across emotional states. The conscious mind continuously produces affect, which serves as an indicator for the body's current state [348]. Affect is the root of emotion [349] but emotion is much more complex, as it classifies affective feelings into more complex categories that often overlap and may only be subtly distinguished [350]. Within software analysis, it can help distinguish between high-energy responses like surprise or anger versus low-energy ones like sadness or disengagement, giving structure to emotional fluctuation during tasks.

4.3.2.5 Broaden-and-Build Theory

Barbara Fredrickson Broaden-and-Build theory [351] proposed that positive emotions do not simply reflect well-being but broaden individuals' cognitive and behavioural repertoires, fostering creativity, flexibility, and resilience. The upward spirals of development, assumes that positive emotions and cognition are mutually supportive and lead to greater prosperity [352] thereby broadening the creative limits of an individual, especially over time as individuals experiencing more positive emotions become more resilient and in turn, they report increased positive emotions [353]. Negative emotions typically trigger fight-or-flight responses that lead to an immediate benefit (i.e. survival) [354]. This is particularly relevant for understanding recovery and problem-solving during complex interface interactions. While negative emotions like anger or fear may narrow attention to immediate problems, positive emotions like joy and engagement enable users to explore, adapt, and persist in task performance. In this framework, emotional states are not just outcomes but functional drivers of interaction quality.

4.4 End to End Process

This section outlines the end-to-end data analysis process (outlined in Figure. 4.2) that follows the design and stimulus exercises outside of iMotions introduced in Section 3. The analysis begins within the iMotions platform, where participant behaviour is recorded, pre-processed, and annotated. Once the initial in-platform processing is complete, the workflow transitions outside iMotions, where exported statistical summaries and time-series datasets are further analysed using custom scripts and visualisation techniques. The following subsections provide detailed insight into each part of this process.

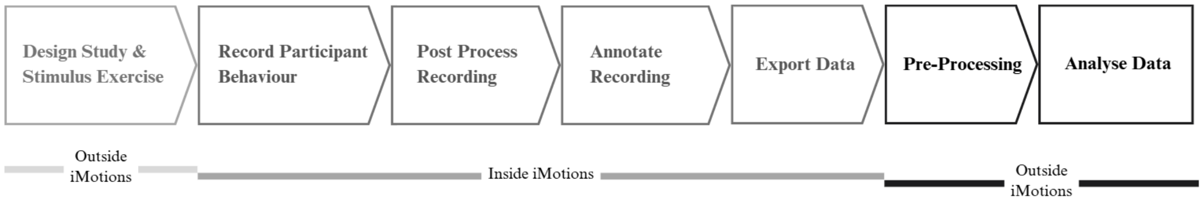


Figure 4.2: Data Analysis Process

4.4.1 Methodology

As the live biometric data was captured simultaneously during the experimental sessions, it follows the same methodological framework and data acquisition procedures detailed in Chapter Three. The data collection process was therefore grounded in the previously defined experimental design, ensuring coherence between the survey-based and biometric components of the study. The biometric recording which occurred during the experimental task were undertaken by participants in both the baseline and modified IDE conditions, aligning with the first two phases of the overall process illustrated in Figure 4.2.This experimental process corresponds directly to Phases 1 and 2 within the end-to-end evaluation pipeline. Presenting it here ensures continuity between the earlier experimental design and the survey-based and biometric analyses that follow, emphasising that the experiment forms an integral component of the overall evaluation procedure.

4.4.1.1 Ethical Considerations

As the dataset includes biometric measures derived from webcam recordings, participants were explicitly informed that facial-expression markers would be analysed as part of the study. Participants retained the right to withdraw from the study in accordance with university ethical procedures, including the removal of their data prior to anonymisation and aggregation where feasible. To protect participant privacy, all biometric records were linked to anonymised participant identifiers rather than personal information. Raw recordings and derived biometric data were stored on secure university-approved systems with access restricted to the research team and managed in accordance with institutional data protection and retention requirements.

4.4.1.2 Environmental Factors

The study was conducted under controlled environmental conditions to ensure consistency across all participant sessions and to minimise external influences on performance, emotional responses, and biometric recordings. All sessions took place in a quiet computing laboratory on level one of the STEAMHouse building with limited foot traffic to reduce auditory and visual distractions. Lighting conditions were kept constant throughout the data collection period using stable indoor artificial lighting to minimise shadows, glare, and illumination variability that could affect facial-expression detection accuracy. Natural lighting variation was minimised by maintaining consistent room conditions during all sessions.

Each workstation operated under the same software and operating conditions to minimise environmental variability. Identical software versions, IDE configurations, operating system settings, and background processes were maintained across all sessions to ensure consistency in system performance, latency, and IDE responsiveness.

Prior to beginning the experiment, participants were given a brief adjustment period to familiarise themselves with the workstation, reducing initial discomfort or unfamiliarity that could influence attention or emotional responses. Sessions were also monitored continuously to identify any environmental or technical disturbances that could affect recording quality.

4.4.1.3 Setup and Calibration

Participants were taken to the study room (illustrated in Figure 4.3) and seated comfortably in front of a 24-inch monitor with a resolution of 1920x1080, equipped with a centrally positioned 1080p webcam mounted above the display. Participants were seated approximately 50–70 cm from the monitor to maintain consistent facial visibility and recording quality across sessions. Prior to data collection, participants were instructed to maintain a relatively stable head position so that their face remained within the camera frame for continuous facial-marker detection. Monitor height and seating position were adjusted where necessary to maintain clear frontal facial visibility and participant comfort. Webcam angle and participant positioning were calibrated before each session to minimise facial distortion, excessive tilt, occlusion, or off-axis positioning. Calibration checks included verification of facial landmark detection stability, camera focus, lighting consistency, frame capture quality, and live video feed reliability before beginning the programming task. Screen brightness and display scaling settings were also standardised across all workstations. The webcam recorded at 1080p resolution using a consistent framerate (30fps) configuration throughout the experiment to support stable facial-feature extraction and emotion analysis.

A trained observer was stationed in the observation room throughout the experiment. They were equipped with a monitor of the same specification, allowing them to monitor participants and their IDE interactions in real time. This setup ensured that behavioural and biometric data were being accurately recorded throughout each session. The observer also monitored for technical issues including facial-tracking instability, software malfunctions, interruptions in the live data stream, or temporary recording failures. To improve dataset reliability, recordings containing prolonged facial loss, excessive participant movement, severe occlusion, or major interruptions to biometric capture quality were flagged during preprocessing for further inspection or exclusion where necessary. This helped maintain consistency and integrity across all recorded sessions while supporting reproducibility of the experimental setup.

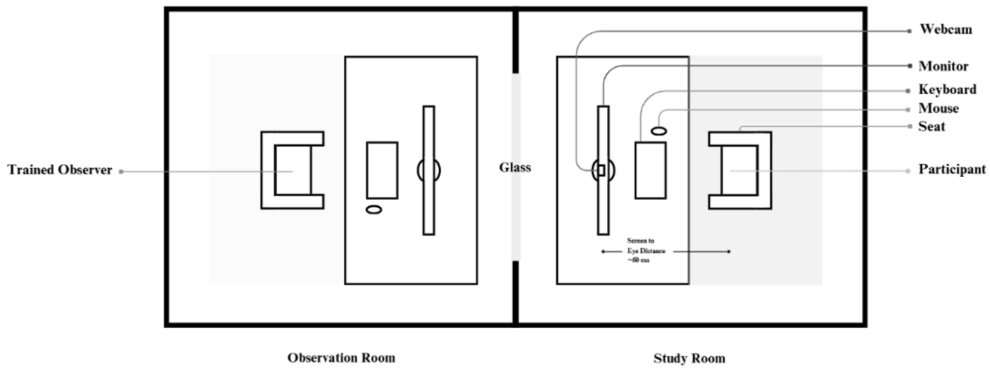


Figure 4.3: Experimental Setup

The setup described in this section is intended to isolate IDE-induced technostress by maintaining a controlled experimental environment in which task structure, time constraints, and interaction conditions were held constant across participants and cycles. By focusing on interaction with the IDE itself, rather than external academic or organisational pressures, the design aims to attribute observed emotional, behavioural, and performance-related responses primarily to IDE features and interaction patterns. However, it is acknowledged that several threats to validity remain (all the internal threats + ET2).

4.4.1.4 Apparatus

The iMotions software v9.4 [355] uses a modular structure, centred around a hub, where numerous biosensors and their respective modules can be incorporated. This comprehensive approach aims to simplify investigations on human behaviour, encompassing study design, stimulus presentation, data collection, and analysis within a single unified software suite. Its purpose is to advance research on human behaviour. As for reliability, iMotions is widely used by 9/10 of the leading universities across the world [356] with hundreds of publications [357].

Facial expression analysis in this study begins with face detection, which is achieved using the Viola–Jones Cascaded Classifier algorithm [358]. This algorithm, commonly used in applications like smartphone cameras, identifies the position of a face within a video frame and highlights it with a bounding box. Once a face is detected, iMotions leverages the Affectiva AFFDEX v5.1 engine [359] to conduct real-time emotion recognition from the webcam feed. Using deep learning and machine learning algorithms, it detects faces and provides probabilities for the facial expressions being made, offering insights into the emotional response at that moment. FEA automatically provides numerical scores for facial expressions, Action Units, and emotions, accompanied by confidence levels. These metrics can be likened to detectors: as facial expressions or emotions occur or intensify; the confidence score increases from 0 (no expression) to 100 (fully expressed). These qualities make it particularly suitable for controlled laboratory studies, such as the present work, where accurate emotional detection, synchronisation, and replicability are essential.

4.4.2 Data Analysis

4.4.2.1 Data Verification

Following the completion of each experimental cycle, all biometric data underwent an internal review conducted by the primary researcher to assess completeness and consistency. Any entries identified as anomalous, such as corrupted or incomplete recordings, were referred to a member of the supervisory team for secondary review. This two-stage verification process helped ensure methodological rigour and maintain data integrity. Coding task correctness was assessed through direct comparison with the task worksheet and predefined test cases, enabling objective determination of requirement achievement.

4.4.2.2 Annotation

Annotations (Figure 4.4) were added to facilitate structured segmentation and focused analysis during the subsequent data-processing stage. The annotation process involved manually identifying and labelling key participant activities and interaction phases throughout each experimental session. Annotated elements included periods when participants were actively interacting with the IDE, engaging in task analysis, implementation, testing activities, and external web usage. Each annotation category was colour coded to support visual interpretation and temporal alignment within the dataset, including IDE usage (red), analysis stage (pink), implementation stage (purple), testing stage (dark blue), and web access periods (light blue).

Phase boundaries were determined based on observable behavioural transitions and interaction patterns within the screen recordings and live monitoring process. The analysis phase typically represented periods where participants were reading, interpreting, or planning solutions before substantial code interaction occurred. The implementation phase represented active code writing or modification, while the testing phase captured compilation, execution, debugging, and validation activities. Web-access annotations identified intervals where participants temporarily shifted attention away from the IDE to external online resources such as documentation, tutorials, or error-resolution searches.

The annotation process was conducted manually by the primary researcher. To improve annotation consistency, predefined annotation criteria and phase definitions were established prior to analysis. Quality checks were performed through repeated review of recordings to verify annotation timing accuracy, phase-transition consistency, and alignment with observable participant actions.

As the annotation and coding process was primarily performed by a single annotator, formal inter-annotator agreement measures were not established. This introduces the possibility of subjective interpretation when determining phase transitions or behavioural categorisation boundaries. To mitigate this limitation, annotation guidelines were applied consistently across all recordings, and ambiguous segments were reviewed iteratively to improve reliability and reduce inconsistency in temporal segmentation.

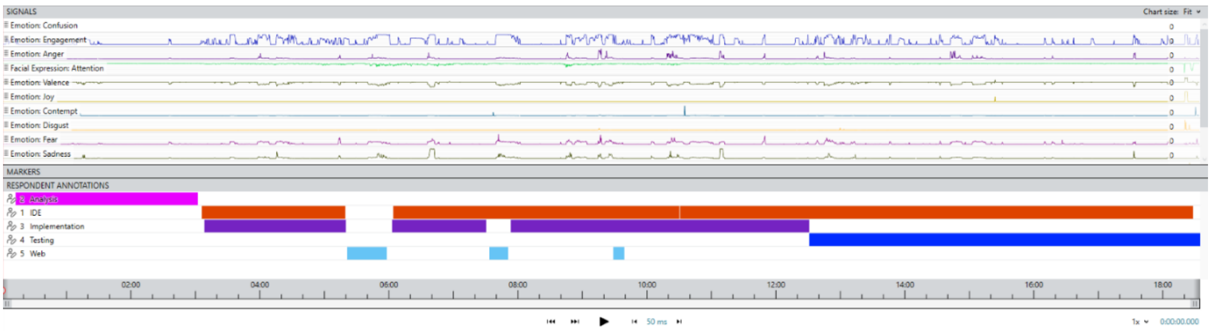


Figure 4.4: Recording Annotation

4.4.2.3 Post-Processing

The recorded data for each participant undergoes post-processing to analyse facial expressions for every frame. All sessions show sub-standard data quality at first, due to AFFEX's initial rendering rather than due to poor facial expressions. To reduce the processing load on the computer, AFFDEX renders at 15 Hz during live collection and must be post-processed up to 30 Hz after recording (Figure 4.5). This metric is only representative of the percentage of frames successfully analysed by iMotions and is not an indicator of overall accuracy or face quality.

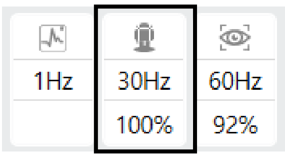


Figure 4.5: Post-Processing Recorded Data

4.4.3 Export

To support reproducibility, modular analysis, and dataset reuse across HCI and affective computing research, all data collected from the study has been compiled into a structured, multi-component dataset titled TIDE. The dataset consists of four interrelated exports (Figure 4.6), each aligned by a common participant ID and collected at different stages of the experimental workflow.

The four components of TIDE are:

(1) TIDE-DM (Demographics Metadata): Contains anonymised participant background data such as gender, ethnicity, academic year, and coding experience. This data was collected through a pre-experiment questionnaire and supports demographic stratification, diversity analysis, and experience-based comparisons.

(2) TIDE-TS (Time-Series): Includes raw, second-by-second biometric data comprising facial emotion intensities and gaze coordinates, collected continuously throughout the programming task. Gaze data can be used to determine which Areas of Interest (AOIs) within the IDE interface (e.g., Editor, Menu, Terminal) were being attended to at different time points.

(3) TIDE-SS (Statistical Summary): Derived from TIDE-TS, this file provides aggregated emotion statistics per participant and phase (Analysis, Implementation, Testing), including mean, standard deviation, and variance for each emotion marker.

(4) TIDE-TO (Task Outcomes): Captures task-level performance metrics collected after task completion, including completion time, number of requirements met, test case success rate, and final task accuracy. This component enables comparison between emotional patterns and measurable productivity or performance outcomes.

Each dataset module is independently useful but can be merged via the shared Participant ID to support integrated, multi-layered analysis.

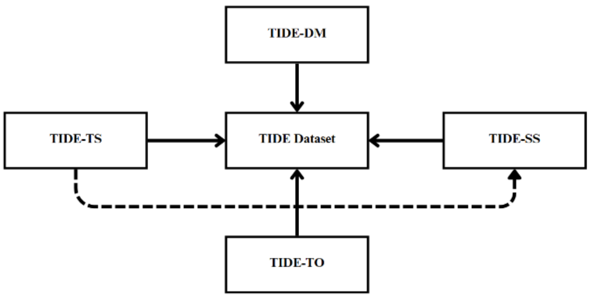


Figure 4.6: Core Dataset and its Sub-datasets

4.4.3.1 Pre-Experiment Data

4.4.3.1.1 TIDE-DM

4.4.3.1.1.1 Dataset Description

Demographic information (Figure. 4.7) collected from participants is compiled into a structured table referred to as TIDE-DM.

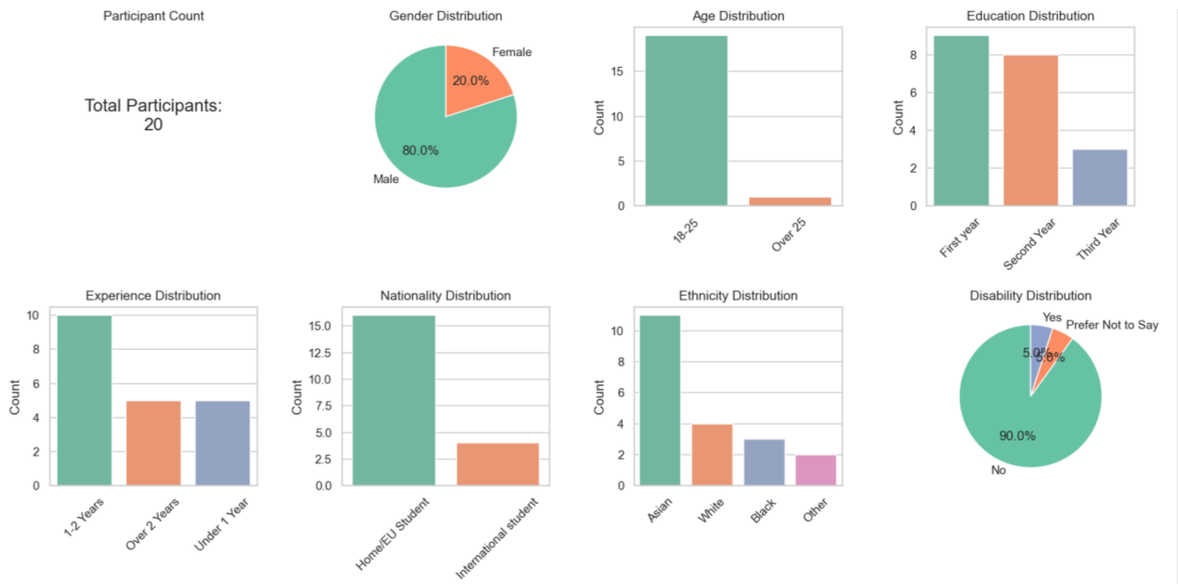


Figure 4.7: TIDE-DM Distribution

4.4.3.1.1.2 Dataset Structure

Each row in the dataset corresponds to a single participant's demographic record (Figure. 4.8). The dataset contains 20 rows and 8 columns, of these 8 all are categorical. This anonymised file captures attributes such as gender, age, year of study, coding experience, nationality, ethnicity, disability and is indexed using a unique participant ID.

P	Gender	Age	Education	Experience	Nationality	Ethnicity	Disability
1	Male	Over 25	Second Year	1-2 Years	Home/EU Student	Black	No
2	Male	18-25	Second Year	Over 2 Years	Home/EU Student	White	No
3	Female	18-25	First year	1-2 Years	Home/EU Student	Asian	No
4	Female	18-25	Second Year	1-2 Years	Home/EU Student	Asian	No
5	Male	18-25	First year	1-2 Years	Home/EU Student	Asian	No

Figure 4.8: TIDE-DM Structure

4.4.3.2 Intra-Experiment Data

4.4.3.2.1 Pre-Processing

Prior to plotting, the data underwent preprocessing and cleaning to ensure quality and consistency. This included handling missing values, verifying data types, and reformatting participant identifiers where needed. To facilitate direct comparison between the two experimental conditions, Cycle 1 (baseline IDE) and Cycle 2 (modified IDE). The datasets from both cycles were concatenated into a unified structure. A new column labelled "Cycle" was added to the merged dataset, with values " 1" and " 2" to clearly distinguish between the conditions.

To prepare the TIDE-TS dataset for analysis (Figure. 4.9), facial markers of interest were first defined. Data from Cycle 1 and Cycle 2 were then loaded, with column names stripped of whitespace and each dataset labelled by its respective cycle. An initial unclean combined dataset was created, after which a cleaned version was generated by retaining only relevant columns (i.e., respondent ID, timestamp, facial markers, and cycle label) and removing any rows with missing facial marker values. This ensured consistency and integrity across both experimental conditions for downstream analysis.

```python
# Facial markers of interest
facial_markers = [
    "Anger", "Contempt", "Disgust", "Fear", "Joy", "Sadness", "Surprise",
    "Engagement", "Valence", "Sentimentality", "Confusion", "Attention"]

# Load and label
df1 = pd.read_csv("TIDE_TS_C1.csv", on_bad_lines='skip')
df2 = pd.read_csv("TIDE_TS_C2.csv", on_bad_lines='skip')
df1.columns = df1.columns.str.strip()
df2.columns = df2.columns.str.strip()
df1["Cycle"] = "1"
df2["Cycle"] = "2"

# Create unclean combined dataset
df_unclean = pd.concat([df1, df2], ignore_index=True)

# Clean dataset, only relevant columns and drop rows with missing facial markers
common_columns = ["Respondent", "Timestamp"] + facial_markers + ["Cycle"]
df1_clean = df1[df1.columns.intersection(common_columns)]
df2_clean = df2[df2.columns.intersection(common_columns)]
df_clean = pd.concat([df1_clean, df2_clean], ignore_index=True)
df_clean = df_clean.dropna(subset=facial_markers)
```

Figure 4.9: TIDE-TS Pre-Processing Code

A similar approach was taken for the TIDE-SS dataset (Figure. 4.10)

```python
# Facial markers of interest
facial_markers = [
    "Anger", "Contempt", "Disgust", "Fear", "Joy", "Sadness", "Surprise",
    "Engagement", "Valence", "Sentimentality", "Confusion", "Attention"]

# Load and label
df1 = pd.read_csv("TIDE_SS_C1.csv", on_bad_lines='skip')
df2 = pd.read_csv("TIDE_SS_C2.csv", on_bad_lines='skip')
df1.columns = df1.columns.str.strip()
df2.columns = df2.columns.str.strip()
df1["Cycle"] = "1"
df2["Cycle"] = "2"

# Combine datasets
df = pd.concat([df1, df2], ignore_index=True)

# Clean and filter
df["Facial Marker"] = df["Facial Marker"].str.strip()
df["Label"] = df["Label"].str.strip()
df["Respondent"] = df["Respondent"].astype(str).str.strip()
```

Figure 4.10: TIDE-SS Pre-Processing Code

4.4.3.2.2 TIDE-TS

4.4.3.2.2.1 Dataset Description

The time-series export preserves the full sequential resolution of biometric and emotional data captured throughout the experiment. Each respondent's data is stored in individual *.csv* files, containing a continuous stream of timestamped measurements recorded during the participant's interaction with the IDE. Each cycle dataset (Figure 4.11) has approximately ~600 thousand rows of data, and 15 columns, of these 2 of which are categorical/objects (respondent and cycle) and 13 are numerical/floats (timestamp and the 12 target emotions).

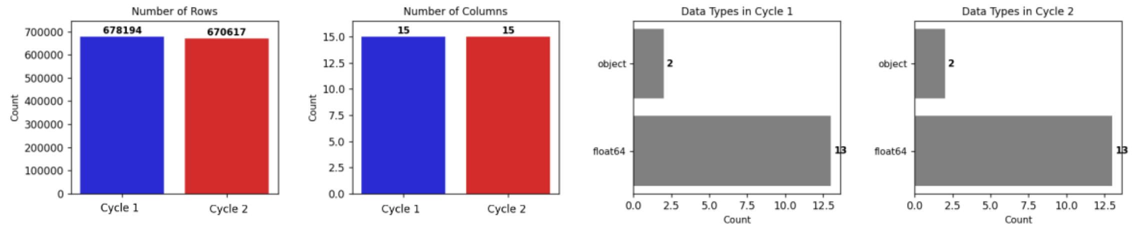


Figure 4.11: TIDE-TS Distribution

Box plots generated (Figure 4.12) compare the average intensity of each target emotion between Cycle 1 and Cycle 2. Each box plot represents the distribution of participant-level means for the given emotion across the two experimental conditions.

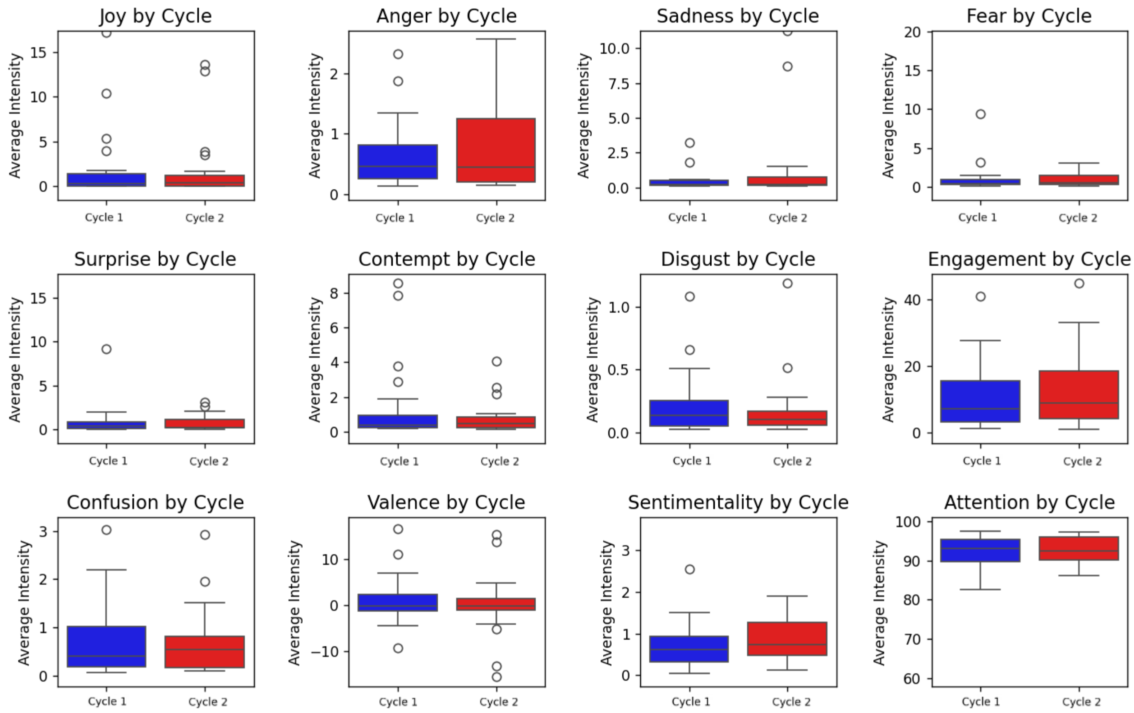


Figure 4.12: TIDE-TS Avg Intensity Across Each Emotion

4.4.3.2.2.2 Dataset Structure

Each row (as seen in Figure. 4.13) in the dataset reflects a moment-by-moment snapshot of the participant's emotional state, while each column represents a distinct biometric variable. iMotions FEA engine dynamically assigns numerical scores based on the intensity and presence of facial expressions and emotions, with values ranging from 0 to 100 for most expressions, and from -100 to +100 in the case of valence.

Respondent	Timestamp	Anger	Contempt	Disgust	Fear	Joy	Sadness	Surprise	Engagement	Valence	Sentimentality	Confusion	Attention	Cycle
	510.0	0.110762	0.126786	0.020385	0.112300	0.048990	0.106814	0.059358	0.328766	0.489830	0.037175	0.037175	98.28521	1
	703.0	0.119230	0.142288	0.021609	0.120833	0.041177	0.112371	0.060802	0.328766	0.000000	0.077294	0.077294	97.11965	1
	708.0	0.108597	0.118371	0.020188	0.109590	0.056889	0.104258	0.060031	0.328766	1.700753	0.091148	0.091148	96.97689	1
	813.0	0.104418	0.105374	0.019634	0.105164	0.069947	0.099354	0.060822	0.328766	3.507161	0.111808	0.111808	97.14111	1
	817.0	0.103527	0.102313	0.020032	0.103886	0.077983	0.098613	0.061202	0.328766	4.264450	0.107820	0.107820	97.28928	1

Figure 4.13:TIDE-TS Structure

4.4.3.2.3 TIDE-SS

4.4.3.2.3.1 Dataset Description

The statistical summary export generates a condensed dataset derived from raw AFFDEX recordings, enabling focused parametric analysis across predefined experimental stages. This summary includes key metrics such as the mean (m), standard deviation (sd), and variance (v) for each facial marker captured during specific stimulus durations. The mean represents the average probability score of a given facial expression within the selected timeframe, while the standard deviation indicates the degree of variation from this average where a lower standard deviation suggests closely clustered values, and a higher value indicates greater variability. Variance further contextualises this spread by measuring how far the data points deviate from the mean across the session.

Each cycle dataset (Figure 4.14) has approximately ~3 thousand rows of data, and 7 columns, of these 4 of which are categorical/objects (respondent, label, facial marker and cycle) and 3 are numerical/floats (m, sd, v).

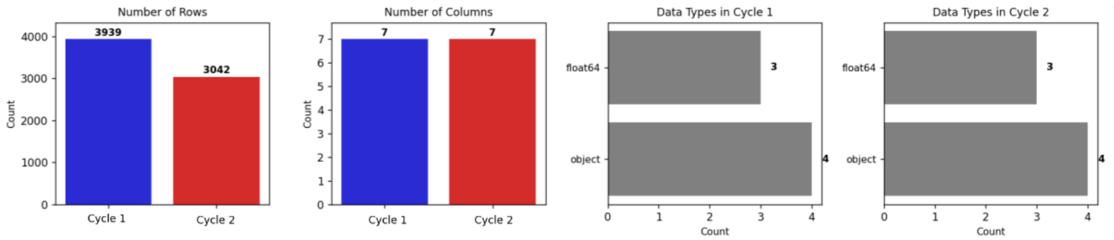


Figure 4.14: TIDE-SS Distribution

4.4.3.2.4 Dataset Structure

The exported dataset (as seen in Figure. 4.15) includes columns denoting the participant's name, the annotated phase (such as analysis, implementation, or testing), the specific facial marker, and the corresponding statistical values. This summary format is useful for comparing aggregated emotional responses across participants and phases. Furthermore, it allows for the generation of mean emotional trendlines across each experimental phase, supporting sequential comparisons between distinct stages of the user experience.

Respondent	Label	Facial Marker	m	sd	v	Cycle
	ScreenRecording-1	Anger	1.344194	1.603341	2.570702e+00	1
	ScreenRecording-1	Contempt	0.337092	2.045899	4.185701e+00	1
	ScreenRecording-1	Disgust	0.028148	0.021300	4.537090e-04	1
	ScreenRecording-1	Fear	3.183039	5.264605	2.771606e+01	1
	ScreenRecording-1	Joy	0.024209	0.000915	8.376870e-07	1

Figure 4.15: TIDE-SS Structure

4.4.3.3 Post-Experiment Data

4.4.3.3.1 TIDE-TO

4.4.3.3.1.1 Dataset Description

The TIDE-TO export (Figure. 4.16) captures individual task outcome metrics aligned with experimental conditions, providing a concise, participant-level overview of performance within each IDE cycle. This dataset includes both binary and scalar indicators of task success, enabling direct comparisons across participants and conditions (baseline vs. modified IDE). It is particularly valuable for linking emotional patterns (from biometric datasets) with concrete task execution outcomes.

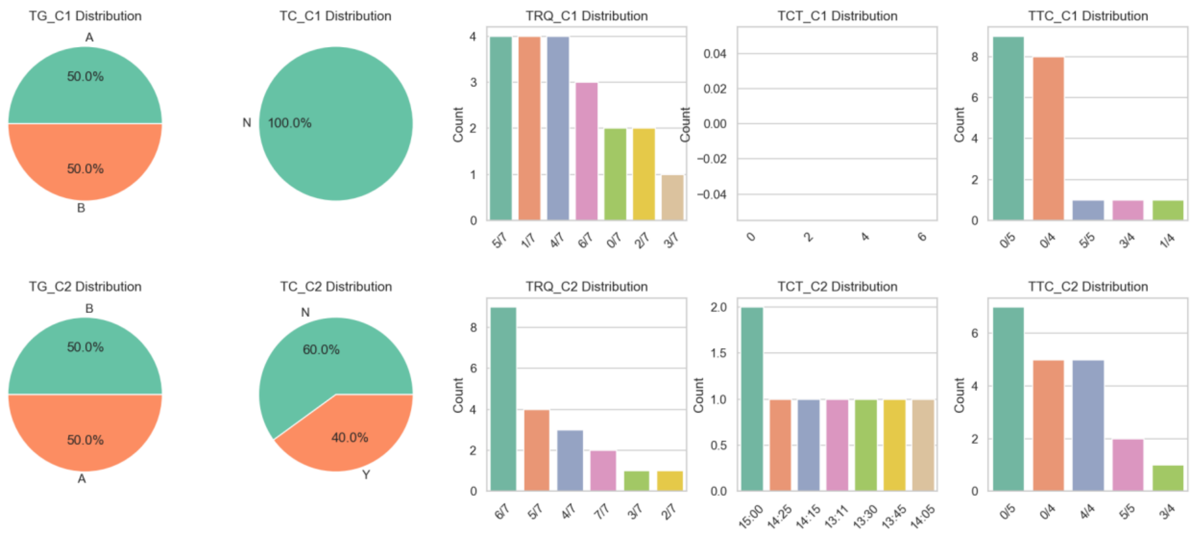


Figure 4.16: TIDE-TO Distribution

4.4.3.3.1.2 Dataset Structure

Each row in the dataset corresponds to a single participant's performance record for a given task cycle (Figure. 4.17). The dataset contains 20 rows and 12 columns. With each row capturing key outcome variables related to how participants interacted with the IDE during the experiment. The key columns are: P (Participant), TG (Task Given), TC (Task Completion), TRQ (Task Requirements), TCT (Task Completion Time), TTC (Task Test Cases), and MV (Modified Version Preference). Columns 2-6 represent Cycle 1, with columns 7-11 representing Cycle 2.

P	TG_C1	TC_C1	TRQ_C1	TCT_C1	TTC_C1	TG_C2	TC_C2	TRQ_C2	TCT_C2	TTC_C2	MV
1	A	N	0/7	NaN	0/4	B	Y	7/7	14:25	5/5	Y
2	B	N	6/7	NaN	5/5	A	Y	7/7	15:00	3/4	Y
3	B	N	3/7	NaN	0/5	A	N	6/7	NaN	0/4	Y
4	A	N	5/7	NaN	0/4	B	N	4/7	NaN	0/5	Y
5	B	N	5/7	NaN	0/5	A	Y	5/7	14:15	4/4	Y

Figure 4.17: TIDE-TO Structure

4.5 Procedure

The combined experimental procedure for chapter Three and Four is illustrated in Figure. 4.18 and is comprised of several structured stages designed to ensure consistency across participants and minimise variability in data collection. Each participant progressed through the steps illustrated. In Cycle 2 steps two and three are bypassed as previous informed consent and sociodemographic are same as before.

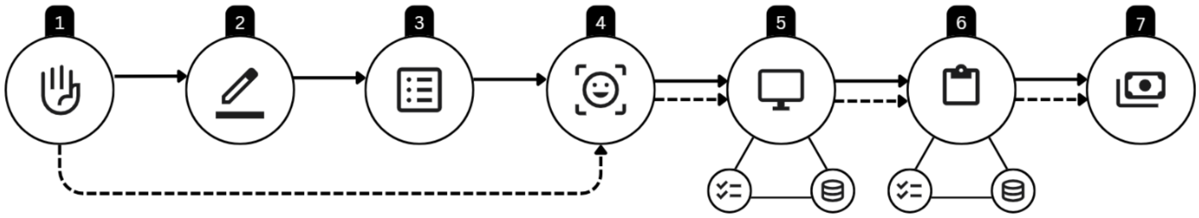


Figure 4.18: Experimental Procedure

4.5.1 Welcome and Briefing

Upon arrival, participants were greeted and given a general overview of the study's aims via a "*participant information sheet*". To minimise bias, the specific hypotheses being tested were not disclosed at this stage. Participants were informed that the study was focused on understanding user experiences and interactions within a programming environment.

4.5.2 Informed Consent

Participants were provided with a detailed informed consent form (Appendix Part A), which outlined the study's objectives, the procedures involved, the voluntary nature of participation, and their right to withdraw at any point without penalty. The form also explained how data would be anonymised, stored securely, and used solely for academic research purposes. Only after confirming understanding and signing the form did participants proceed to the next step. The participant information sheet and consent form used in this study can be found as part of the supplementary material.

4.5.3 Pre-Experiment Questions

Prior to the experimental task, participants completed a short questionnaire capturing key sociodemographic information, including age, gender, educational background, programming experience, and familiarity with IDEs. This data was used to contextualise the results and enable post-hoc analysis.

4.5.4 Task Execution

During the task execution phase, participants were instructed to attempt the given programming task within a controlled environment. Throughout the task, live biometric data, primarily emotion-based facial metrics was continuously captured using the afore mentioned recording setup. All task-related outputs, including the written code and user interactions within the IDE, were stored locally on the testing machine to ensure data integrity. Upon task completion, each solution was systematically verified against a predefined set of requirements and test cases of the given task to assess the correctness and completeness of the implementation.

4.5.5 Post-Experiment Surveys

After completing the experimental tasks, participants were instructed to fill out both the qualitative and quantitative surveys, as outlined in (Chapter 3.2). Once participants had submitted their responses, they were informed that their submissions would undergo a quality assurance review to ensure completeness, consistency, and validity of the data. This review process was not evaluative of participants' individual performance but rather served to maintain the integrity and reliability of the dataset, confirming that all responses met the required standards for inclusion in subsequent analyses.

4.5.6 Debrief and Reward

Finally, participants were thanked for their time and valuable contribution to the study. They were informed that their participation would involve a follow-up session in a subsequent cycle. In recognition of their involvement, each participant received a voucher as compensation for their time. Additionally, participants were provided with contact details for the research team, should they wish to receive updates on the study's outcomes or request the withdrawal of their data at any point.

4.6 Results

This section presents the key findings derived from the multi-layered analysis of the TIDE dataset. By comparing emotion data across Cycle 1 (standard IDE) and Cycle 2 (modified IDE), uncovering how interface design affects emotional dynamics, performance outcomes, and user experience. These findings aim to address RQ1-4.

4.6.1 Emotional Analysis Across Phases and Cycles

The correlation matrix heatmap (Figure 4.19) generated using TIDE-TS contextualises how emotional indicators interact within each cycle. Since correlation reflects the strength and direction of the association between two emotional markers, this figure offers a preliminary view of emotional interdependencies before deeper analysis. These relationships are examined in greater detail in this section as part of the research questions on emotion–emotion correlation.

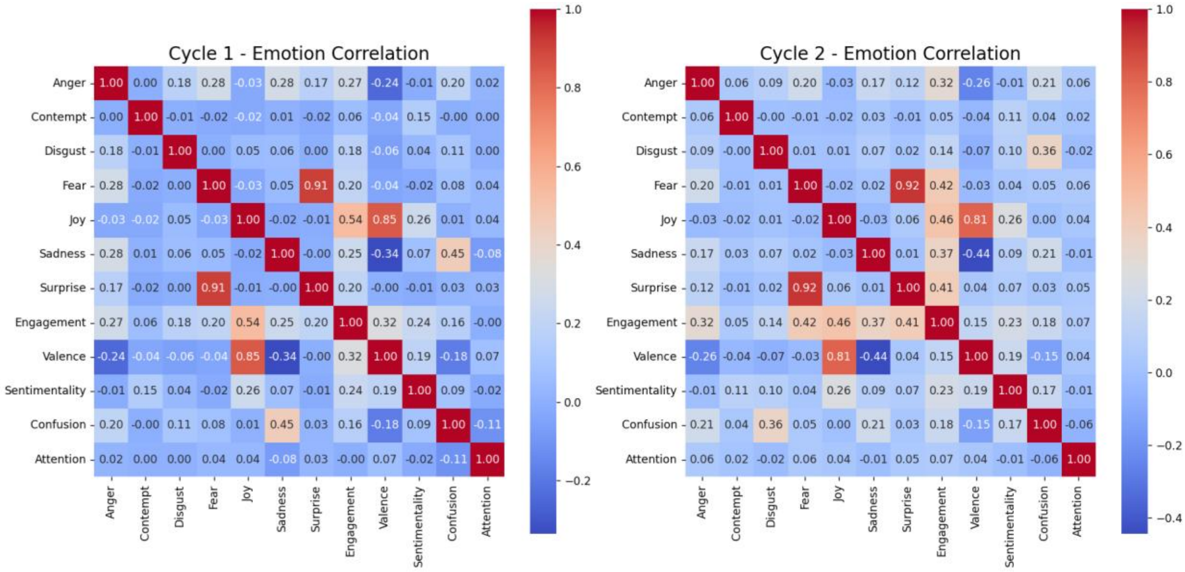


Figure 4.19: Emotion Correlation Matrix Between Cycles

Using TIDE-SS the line plots (Figure 4.20) illustrate the mean intensities of twelve facial markers across three task phases: Analysis, Implementation, and Testing. These visual trends offer an initial indication of how emotions fluctuate throughout the programming workflow, with a fuller interpretation of phase-based emotional dynamics developed later.

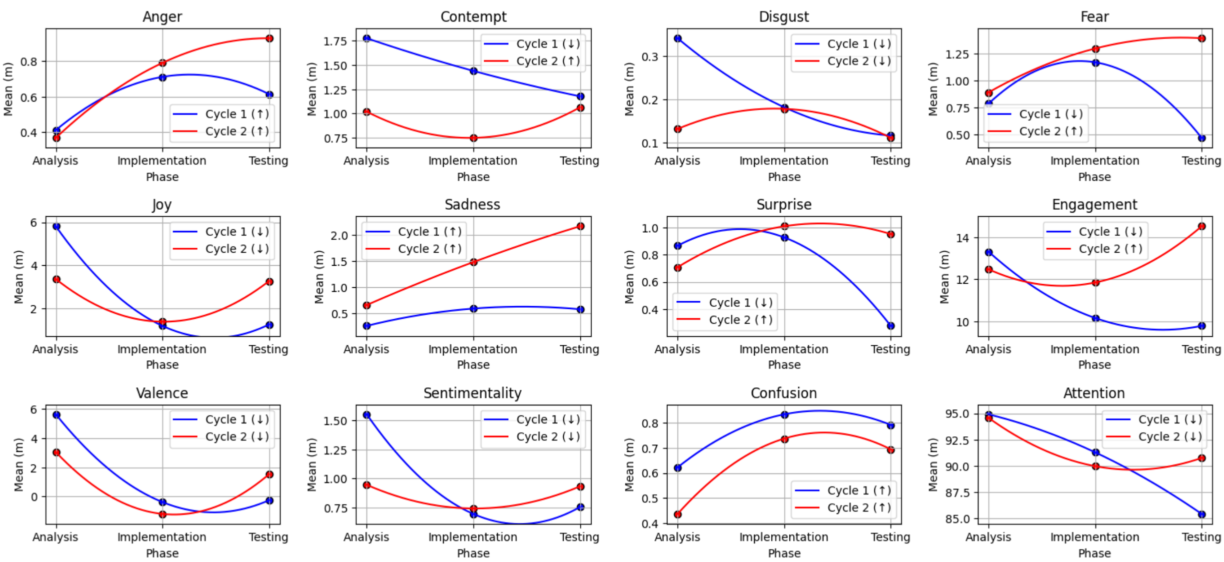


Figure 4.20: Shifts in Target Emotional Markers Across Phases

4.6.1.1 Anger

Anger increased across all phases in both cycles but with distinct trajectories. In Cycle 1 anger rose sharply from Analysis to Implementation, then slightly declined during Testing, a pattern consistent with initial cognitive friction followed by disengagement. In contrast, Cycle 2 saw anger steadily increase, peaking during Testing. Drawing on Lazarus' Appraisal Theory, this sustained rise may not signify frustration but a goal-relevant activation, users likely perceived the task as challenging yet achievable, leading to constructive arousal rather than emotional breakdown. From Pekrun's Control-Value Theory, anger during Testing in Cycle 2 may indicate task investment under pressure, reflecting heightened value perception and partial control, especially as task completion rates were higher. According to Ekman, anger is a basic emotion often tied to obstacles. However, in the presence of rising engagement and joy (Cycle 2), its co-occurrence suggests focused determination.

Correlation*: Anger correlated moderately with engagement (r = 0.30), supporting the interpretation that in Cycle 2, anger reflected focused frustration or task-oriented arousal rather than alienation. Anger's inverse correlation with valence (r = –0.25) also reinforces its role as a challenge-related emotion, dampening positivity when excessive.*

4.6.1.2 Contempt

Contempt in Cycle 1 began relatively high and decreased across phases, likely indicating initial user disapproval or scepticism toward the IDE's interface. This aligns with Ekman's theory, in which contempt represents moral or evaluative rejection. Its decrease may signal adaptation or resignation. In Cycle 2, contempt started lower, dropped slightly during Implementation, but rebounded during Testing, a less stable pattern but with overall lower magnitude across all phases. From a cognitive appraisal lens, the initial low contempt in Cycle 2 may reflect a more favourable first impression or higher perceived affordance [360]. The Testing-phase rebound could relate to increased emotional intensity near completion, especially under time pressure.

Correlation*: While contempt wasn't among the top correlating emotions in the matrix, its trends loosely mirror those of disgust and confusion, which both cluster in contexts of interface conflict or uncertainty. Its independence in correlation supports the interpretation of early evaluative disapproval rather than reactive affect.*

4.6.1.3 Disgust

Disgust decreased across all phases in both cycles but was significantly higher in Cycle 1, especially during Analysis. This pattern suggests that the baseline IDE triggered an aversive reaction. In Cycle 2, disgust remained low and relatively flat, evidence of improved usability and better first impressions. Using Ekman's model, disgust represents a basic rejection emotion, often triggered by perceived violation or cognitive overload. From Russell's perspective, it's a low-valence, high-arousal state, suggesting early-stage withdrawal or tension. Its decline in Cycle 1 may reflect either adjustment or disengagement, while its low levels in Cycle 2 suggest design-level mitigation of negative first impressions.

Correlation*: Disgust clusters with confusion in the correlation matrix, both typically emerge under interface ambiguity or cognitive dissonance, supporting their joint decline as a marker of reduced UI friction in Cycle 2.*

4.6.1.4 Fear

Fear increased from Analysis to Implementation in both cycles, a likely reflection of growing task demand and uncertainty. However, the cycles diverged in Testing: in Cycle 1, fear dropped sharply (coinciding with low participant continuation), while in Cycle 2, it continued to rise, even as participants remained engaged. According to Lazarus, this pattern indicates that in Cycle 2, fear coexisted with perceived coping ability, a functional anxiety driving performance. Pekrun's theory would classify this as a negative activating emotion under high-value, moderate-control conditions. The continued fear in Cycle 2 alongside rising engagement supports this interpretation. Importantly, fear did not deter engagement, implying a productive arousal state.

Correlation*: Fear correlated very strongly with surprise (r = 0.92), suggesting shared cognitive arousal components, both track alertness and uncertainty. Fear also had a moderate correlation with engagement (r = 0.34), reinforcing that fear co-occurred with active involvement rather than withdrawal, particularly in Cycle 2.*

4.6.1.5 Joy

Joy followed a downward trajectory in both cycles during Implementation but only recovered in Cycle 2 during Testing. In Cycle 1, joy dropped from a high in Analysis to flat, low levels thereafter suggesting rapid decline in affective rewards. In Cycle 2, the U-shaped pattern (recovery in Testing) aligns with increased task completion and user satisfaction. Drawing from Fredrickson's Broaden-and-Build Theory, this late-stage joy in Cycle 2 may reflect the positive feedback loop from visible success, which can broaden cognitive scope and enhance resilience. From Russell's model, joy occupies the high-valence, high-arousal quadrant, critical for motivated engagement. Pekrun would frame this as a positive activating emotion, more accessible when control and success are perceived.

***Correlation**: Joy strongly correlates with valence (r = 0.83) and engagement (r = 0.50) validating that it is a robust marker of positive task experience. Its alignment with these indicators supports using it as a representation for overall positive affect.*

4.6.1.6 Sadness

Sadness increased across all phases in both cycles but presented different affective implications. In Cycle 1, sadness rose modestly from Analysis to Implementation and plateaued in Testing likely reflecting emotional fatigue or dissatisfaction. In Cycle 2, sadness increased more steeply, peaking during Testing. However, this was accompanied by higher engagement and performance, suggesting a more emotionally invested experience. According to Pekrun, sadness is a negative deactivating emotion, usually signalling disengagement. However, when coupled with continued effort, it may signify reflective involvement or concern over task outcome consistent with Lazarus' appraisal model where sadness stems from goal-blocking or anticipated failure. This interpretation is supported by user behaviour in Cycle 2: users appeared emotionally present and committed despite rising sadness.

***Correlation**: Sadness shows a moderate positive correlation with joy (r = 0.32), an unexpected finding that may reflect emotional richness or complex affective states, particularly under pressure. It also shows a negative correlation with valence (r = –0.39), aligning with the idea that higher sadness reflects reduced emotional positivity, especially in Cycle 1.*

4.6.1.7 Surprise

Surprise declined across phases in Cycle 1, suggesting initial novelty wore off, possibly due to low engagement or a lack of meaningful interface affordances. In Cycle 2, surprise increased through Implementation and remained high during Testing, indicating sustained cognitive stimulation and possibly exploratory behaviour as users engaged with modified IDE. According to Russell, surprise occupies a high-arousal, ambiguous-valence zone, often linked to novelty and attention redirection. Lazarus would view it as resulting from unexpected appraisal outcomes. In Cycle 2, the stability of surprise may reflect the interface's ability to retain user curiosity and interaction depth.

***Correlation**: Surprise shows an extremely strong positive correlation with fear (r = 0.92), indicating that both tap into cognitive alertness and arousal, supporting the idea that sustained surprise in Cycle 2 represents engaged mental effort, not confusion.*

4.6.1.8 Engagement

Engagement steadily declined in Cycle 1, especially from Implementation to Testing, suggesting cognitive fatigue or interface failure to sustain motivation. In Cycle 2, engagement dropped slightly during Implementation but rebounded in Testing, reflecting renewed investment as users approached completion and potentially perceived greater control or reward. Fredrickson's theory supports the notion that positive affect fosters engagement, while Pekrun's framework places engagement as a result of perceived control and value. The upward trend in Cycle 2's Testing phase suggests that the modified IDE better maintained motivational structures. Lazarus would explain this as the result of appraisals that the task remains relevant and manageable.

***Correlation**: Engagement correlates positively with joy (r = 0.50) and fear (r = 0.34), suggesting it arises both from positive reward and task-related arousal. Its moderate correlation with anger (r = 0.30) further reinforces the idea that emotional activation (not just positivity) predicts engagement.*

4.6.1.9 Valence

Valence, a general marker of emotional positivity, dropped during Implementation in both cycles. In Cycle 1, it remained low into Testing, whereas in Cycle 2, valence partially recovered, mirroring increases in joy and engagement. This rebound implies higher satisfaction and perceived progress under the modified IDE. According to Russell's circumplex model, valence is a foundational emotional dimension, and it's rebound in Cycle 2 signals a shift toward more favourable emotional state. Lazarus would attribute this to reappraisal of the task as less threatening and more rewarding in later stages.

***Correlation**: Valence is strongly correlated with joy (r = 0.83) and negatively with sadness (r = –0.39) and anger (r = –0.25) supporting its use as a global emotional indicator and confirming that emotional improvements in Cycle 2 were substantial.*

4.6.1.10 Sentimentality

Sentimentality declined in both cycles, but remained higher in Cycle 2, especially in Testing. This suggests that users in Cycle 2 developed a more personal or reflective connection to the task as a result if the modified IDE. Although often overlooked in standard emotional taxonomies, sentimentality can be tied to Fredrickson's concept of positive meaning-making, especially under prolonged effort. It may also indicate a longer-term emotional investment, potentially evoked by a sense of accomplishment in the modified IDE experience. Lazarus' theory supports this by explaining sentimentality as a secondary appraisal reflecting emotional memory or anticipated loss (e.g., the task ending).

Correlation*: Sentimentality correlates weakly with joy (r = 0.26), suggesting its presence in moments of fulfilment or reflection, especially under successful performance conditions as seen in Cycle 2.*

4.6.1.11 Confusion

Confusion increased slightly across phases in both cycles, but was markedly higher in Cycle 1, especially during Implementation. This suggests that the baseline IDE introduced cognitive ambiguity or interface misalignment. In Cycle 2, confusion remained lower, suggesting improved user experience. According to Russell, confusion is high arousal but varies in valence depending on context. From Lazarus' model, confusion results from appraisal conflict, unclear information paired with task relevance. Its lower prevalence in Cycle 2 supports the notion of design-driven clarity.

Correlation*: Confusion clusters with disgust in the correlation matrix, further supporting their shared origins in user interface friction and cognitive overload.*

4.6.1.12 Attention

Attention declined in both cycles over time, a typical cognitive pattern under load. However, in Cycle 1, the decline was steeper, especially into Testing. In Cycle 2, attention dropped more gradually and remained higher overall, suggesting greater persistence and cognitive investment, likely due to modifications implemented and task engagement. Fredrickson's model suggests that positive affect (e.g., joy, satisfaction) sustains attention by expanding cognitive scope. Pekrun emphasises that perceived value and control are crucial for sustained effort. The shallower decline in Cycle 2 suggests greater resilience and task interest.

Correlation*: Although not highly correlated with individual markers, attention tracks with the general trend of engagement and valence, supporting its role as an indirect measure of emotional alignment with task progress.*

4.6.1.13 PCA

As for PCA (Figure. 4.21) PC1 (x-axis) explains 20.3% of the total variance in the dataset. PC2 (y-axis) explains 18.0% of the variance. Together, these two components explain 38.3% of the variation in the data. The plot shows a funnel-like distribution, with most data points (especially from Cycle 1) densely concentrated toward the centre and spreading outward as PC1 increases. Cycle 2 (red) is similarly concentrated near the centre but less so across the PC1 axis, implying greater variance and more diversity in emotional responses.

Cycle 1 appears more constrained, which implies a more consistent but less dynamic user experience and a less emotionally responsive system (e.g., baseline IDE not stimulating strong reactions). Cycle 2 which exhibits greater emotional variability, indicates a richer or more intense user experience as a result of greater engagement, stronger emotional reactions and a wider range of responses due to changes in the modified IDE.

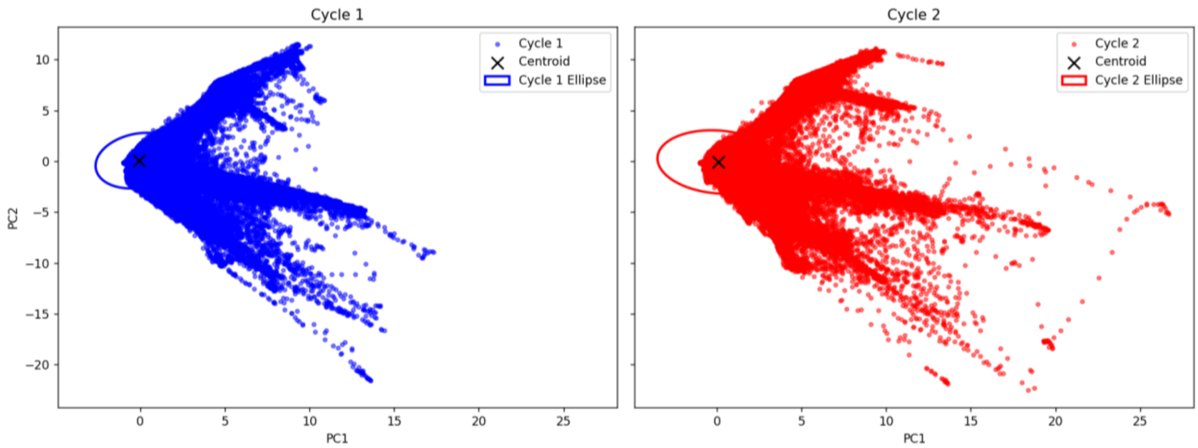


Figure 4.21: PCA Projection Cycle 1 vs Cycle 2

4.6.1.14 MANOVA

A Multivariate Analysis of Variance (MANOVA) was conducted to examine whether the set of twelve facial emotion markers differed significantly between participants in Cycle 1 (baseline IDE) and Cycle 2 (modified IDE). The resulting output includes two main test tables (Figure. 4.22): Intercept and Cycle. The Intercept tests whether the overall emotion means differ from zero, this is standard in such models but is not central to interpretation here. The Cycle table, however, is of primary interest, as it assesses whether the combined emotional profiles differed significantly between the two cycles.

The MANOVA revealed a statistically significant multivariate effect of cycle on facial emotion responses, as indicated by Wilks' Lambda = 0.9839, F(12, 1,348,798) = 1839.80, $p < .001$. This result was confirmed by all other multivariate test statistics, including Pillai's Trace = 0.0161, Hotelling's Trace = 0.0164, and Roy's Greatest Root = 0.0164, all significant at $p < .001$. These findings indicate that participants' emotional profiles differed systematically across the two experimental cycles. Despite the statistical significance, the effect size is small, with Pillai's Trace suggesting that approximately 1.6% of the total variance in the twelve emotion markers can be attributed to differences between the two cycles. This is not uncommon in large-scale biometric datasets, where even modest shifts in affective state across groups can carry meaningful implications.

Intercept	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.0448	12.0000	1348798.0000	2394713.3413	0.0000
Pillai's trace	0.9552	12.0000	1348798.0000	2394713.3413	0.0000
Hotelling-Lawley trace	21.3053	12.0000	1348798.0000	2394713.3413	0.0000
Roy's greatest root	21.3053	12.0000	1348798.0000	2394713.3413	0.0000

Cycle	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9839	12.0000	1348798.0000	1839.7973	0.0000
Pillai's trace	0.0161	12.0000	1348798.0000	1839.7973	0.0000
Hotelling-Lawley trace	0.0164	12.0000	1348798.0000	1839.7973	0.0000
Roy's greatest root	0.0164	12.0000	1348798.0000	1839.7973	0.0000

Figure 4.22: MANOVA Analysis

4.7 Discussion

4.7.1 Addressing Research Question One

Question: What datasets aid in decision making regarding technostress? (RQ1)

Findings: While numerous studies have explored the psychological and physiological dimensions of technostress, our review found no publicly available datasets that combine biometric emotion signals, gaze behaviour, task performance metrics, and demographics within a structured experimental context involving real-world tool usage. The absence of such a dataset has been a critical barrier for researchers seeking to develop or evaluate technostress-aware systems in applied domains such as education, software engineering, or office productivity.

Discussion: We present TIDE, a novel, multi-modal dataset designed specifically to address this gap. This dataset enables multi-layered, reproducible analysis pipelines and supports both theory-driven and exploratory approaches to understanding technostress in digital environments.

4.7.2 Addressing Research Question Two

Question: Which emotions are representative of others with regards to expressing technostress? (RQ2)

Findings: The correlation analysis and contextual relevance of each emotional marker, provide a more streamlined set of emotions that can be adopted for future studies without compromising analytical depth. Emotions such as Joy, Surprise, and Engagement emerged as key indicators due to their strong correlations with other markers and their relevance to positive affect, cognitive arousal, and motivational states. Joy can effectively represent both Valence and Engagement, given their strong interconnections, while Surprise, closely tied to Fear, can capture emotional alertness and responsiveness to task complexity. Between Disgust and Confusion, only one is necessary. Confusion may be more informative in the context of interface ambiguity and should be prioritised if cognitive dissonance is a focus. Anger and Sadness, while not highly correlated with other emotions, offer important contextual insights, such as frustration, perseverance, or emotional investment, and may be included based on specific research goals. On the other hand, Contempt, Sentimentality, and Valence (already represented by Joy) showed limited standalone utility or redundancy and can be excluded in future iterations. Attention, although cognitive rather than emotional, remains valuable for tracking sustained engagement and should be retained.

Discussion: Therefore, the recommended set of emotions to move forward with includes Joy, Surprise, Engagement, Confusion, Anger, Sadness, and Attention, striking a balance between comprehensive emotional tracking and analytical efficiency.

4.7.3 Linking Findings to Affective-Computing

The observed biometric and emotional patterns also align with broader findings within AC literature, which emphasises that emotions in cognitively demanding environments are often complex, transitional, and context dependent rather than strictly positive or negative. Research by [361] states that confusion, frustration, curiosity, and engagement frequently coexist during effortful problem-solving activities, suggesting that negative-affect indicators do not necessarily correspond to disengagement or failure, but may instead reflect active cognitive processing and learning progression. Similarly, [362] found that novice programmers commonly experienced recurring transitions between confusion, frustration, and engagement during programming activities, particularly during debugging and code-construction tasks.

This interpretation is reflected in the current findings, where emotions associated with cognitive difficulty frequently appeared alongside high attention and engagement levels. Such patterns support theories within AC suggesting that productive struggle is a natural component of complex learning and analytical tasks. In programming contexts specifically, confusion may indicate deep cognitive involvement as participants attempt to resolve syntax errors, logic issues, or unfamiliar concepts rather than simply reflecting usability failure. Research by [363] on "optimal confusion" within interactive digital learning environments similarly argues that moderate levels of confusion can positively contribute to learning and problem-solving when appropriately managed.

The findings also correspond with studies in developer emotion analysis and software-engineering affective research, which report that programming activities often generate rapid emotional fluctuations due to iterative cycles of failure, experimentation, correction, and reward. As [364] highlighted, AC methods are increasingly applicable to understanding programmer behaviour, particularly when combining behavioural and biometric signals during coding activities. For example, debugging activities may simultaneously trigger frustration-related expressions while maintaining high levels of concentration and persistence. Similarly, moments of successful code execution may produce increased joy or engagement following prolonged cognitive effort.

At the same time, the results highlight limitations frequently discussed in AC literature regarding FE interpretation. General-purpose emotion-recognition systems are often trained using universal emotional categories that may not fully capture nuanced cognitive-affective states associated with programming, such as analytical reasoning, mental workload, uncertainty, or sustained concentration [365]. Consequently, certain detected emotional states may reflect broader cognitive strain or task engagement rather than discrete emotional experiences in the traditional psychological sense.

These findings reinforce the importance of multimodal interpretation within affective-computing systems. Facial expressions alone may provide incomplete insight into developer experience unless combined with contextual task information, gaze behaviour, interaction logs, and performance data. The TIDE dataset therefore contributes not only as a biometric dataset for technostress research, but also as an affective-computing resource capable of supporting more context-aware emotional modelling within software-engineering environments.

4.8 Threats to Validity

4.8.1 Internal

The study design prioritised consistency and control, but this also introduced some internal limitations:

IT1. ***Task Scope***: The programming task was time constrained and predefined, which may not reflect the complexity or unpredictability of real-world software development. Emotional reactions might differ in open-ended or collaborative environments.

IT2. ***Tool Environment***: All interactions took place within an IDE environment. Any emotional responses observed are therefore bound to that specific tool, which may limit insight into more diverse software environments.

IT3. ***Observation Bias***: Participants awareness of being recorded may introduce behavioural bias (Hawthorne effect [366]), potentially altering natural expressive behaviour, consciously or unconsciously.

IT4. ***Time Limitation***: Emotional data was captured during a set period duration. It's unclear how affective responses might evolve over longer programming sessions, where fatigue or sustained attention may play a greater role.

IT5. ***Facial Expression Interpretation***: The study relied on automated facial-expression analysis to infer emotional states. However, facial expressions do not always directly correspond to internal emotional experiences. Participants may suppress, exaggerate, or display ambiguous expressions, meaning some detected emotions may not fully represent genuine affective states. Additionally, subtle emotions such as confusion, cognitive overload, or frustration can overlap facially, increasing the possibility of interpretive ambiguity.

IT6. ***Algorithmic Interpretation***: The study relies on automated FEA as a behavioural proxy for emotional states. However, FEs do not always correspond directly to internal emotional experiences, and may be influenced by individual, cultural, and contextual variation. Differences in demographic characteristics such as age, gender, and ethnicity may also affect expression patterns and potentially influence algorithmic interpretation accuracy due to variability in training data representation. In addition, environmental factors such as lighting conditions, camera angle, and recording quality may impact detection reliability. From a theoretical perspective, constructed-emotion accounts further challenge the assumption that discrete facial expressions map consistently to specific emotional states. As a result, FE outputs should be interpreted as a complementary affective signal rather than a definitive measure of experienced emotion or technostress.

IT7. ***Algorithmic Accuracy and Detection Limitations***: The emotion-recognition system used in this study is dependent on machine-learning algorithms that may vary in accuracy under different conditions. Factors such as lighting, camera angle, facial occlusion, head movement, image resolution, and participant positioning may affect detection quality and emotional classification accuracy. As a result, some emotional states may have been misclassified or under-detected.

4.8.2 External

Several aspects limit how broadly the findings and dataset can be generalised:

ET1. ***Participant Pool***: All participants were undergraduate computing students. This restricts the dataset's applicability to other populations such as industry professionals, postgraduate students or students from non-technical disciplines. Therefore, the results most directly reflect IDE-induced technostress as experienced in educational programming contexts rather than professional industrial settings. Although students represent a highly relevant population due to their intensive and sustained IDE usage, differences in experience, responsibility, and workplace pressure mean that direct transferability to professional developers should be approached with caution.

ET2. ***Setting***: The task was conducted in a controlled academic setting. Emotional responses in more dynamic or stressful environments (e.g., team-based software development or time-critical deployments) may differ.

ET3. ***Generalisability***: Although the dataset was collected in the context of an IDE, many of the emotional markers observed are common to a wide range of digital tools. However, further testing is needed before applying these findings beyond IDE-based workflows. For broader generalisability, future experiments should include different digital environments and diverse user groups. Doing so would help confirm whether the emotional patterns found here also apply in other software settings.

ET4. ***Cultural Variation in Emotional Expression***: Emotional expression and interpretation can vary across cultural and social backgrounds. Facial behaviours associated with emotions such as frustration, confusion, or engagement may not be expressed uniformly across all participant groups. Consequently, emotional classifications derived from facial-expression analysis may not generalise equally across culturally diverse populations.

ET5. ***Demographic and Algorithmic Bias***: Emotion-recognition technologies may exhibit variations in accuracy across demographic groups, including differences related to ethnicity, gender, age, facial structure, or appearance. Such biases may influence how emotions are detected or classified, potentially affecting the consistency and fairness of the dataset. Although the study maintained a consistent experimental setup, the possibility of demographic bias within the underlying recognition algorithms cannot be fully excluded.

ET6. ***Transferability of Emotion Recognition Models***: The emotion-detection models used in this study were developed and trained on broader FE datasets that may not specifically represent programming-related cognitive and emotional states. As a result, emotional patterns associated with software development activities may not always align perfectly with general-purpose emotion-recognition frameworks, potentially limiting transferability to other contexts or specialised developer populations.

4.9 Summary

Although previous studies have investigated various negative impacts of technostress, the influence of emotion has predominantly been overlooked [367]. Emotions are a "*mental state of readiness for action that promotes behavioural activation*" greatly influencing human ideas and behaviours and guiding their cognitive processes and decision-making [368]. This calls for complementing the self-reported aspects of technostress (which was the focus of Chapter 3) with the emotional aspects of technostress, using biometric measures. Software development is not only a technical process but also an emotional one. Individuals frequently encounter a mix of positive and negative emotions while solving problems, debugging errors, or meeting deadlines. Emotional experiences during software development can impact productivity, performance, and overall satisfaction [369], [370], [371], [372], [373] positive emotions often propel progress [374], [375] whereas negative ones can hinder it [376].

However, this chapter helps inform technostress-aware design by introducing TIDE as a structured, novel, and multimodal dataset (Obj 1) capturing real-time emotional responses of student developers as they engage with programming tasks in an IDE setting, providing a foundation for future research into technostress-aware user interface design. Additionally, it provides insight into how emotions fluctuate across task phases and tool design cycles, and how these emotional responses relate to task outcomes (Obj 2). Together, the findings and dataset inform both theoretical understanding and the practical evaluation of emotionally supportive digital tools.

Beyond analysis, TIDE enables comparative and benchmark-driven evaluation of IDE design adjustments. For example, TIDE-TS can be used as a reference baseline when evaluating the impact of a specific IDE modification, such as a redesigned menu structure, altered syntax highlighting, or reduced configuration complexity. Subsequent experiments, conducted with new participant cohorts or alternative IDE configurations, can be compared against the emotional and task-outcome patterns observed in TIDE to assess whether a proposed adjustment meaningfully reduces technostress or introduces unintended cognitive or emotional load. In this way, TIDE supports evidence-based decisions about whether particular IDE changes are worth further investment, and whether newly collected biometric data aligns with established emotional patterns or reflects noise or design regression.

The analysis of this chapter revealed key insights. First, TIDE provides a novel and structured dataset to support technostress-related decision-making in digital environments, where such resources remain scarce (RQ1). Second, a reduced but representative subset of emotional markers, including Joy, Surprise, Engagement, Confusion, Anger, Sadness, and Attention, were identified as analytically efficient for future use (RQ2), enabling more focused and scalable evaluation of IDE-induced emotional responses.

Building on the structured construction of the TIDE dataset presented in previous chapter, the proceeding Chapter 5 focuses on the evaluation of both the proposed framework (chapter 3) and the dataset (chapter 3). This evaluation is conducted through expert and user perspectives, intended to further examines how effectively the proposed approach supports technostress-aware IDE analysis and informs future design and research directions.

Chapter *Five*

5 Evaluation

5.1 Introduction

The evaluation in this chapter draws on responses from two distinct groups: industry developers and MSc Computing students (from SoC within BCU) specialising in User Experience Design. These participants not only offer valuable perspectives from professional and academic contexts but also represent the primary users of the framework, who will ultimately employ it to design and develop technostress -aware IDEs capable of identifying and reducing technostress. Their insights provide an essential understanding of how the framework and dataset can be practically integrated into real-world development workflows and educational environments.

In doing so, this chapter provides a dual purpose. First, it provides preliminary expert assessment of the significance of the framework as a foundation for future IDE enhancement strategies that embed technostress awareness into the development process. Second, it establishes the TIDE dataset as a context-specific empirical resource, providing detailed insight into emotional and technostress-related responses within a controlled IDE-based programming setting. While the dataset, it offers a transparent and well-documented basis for interpreting technostress dynamics in relation to task phases, design cycles, and measured outcomes within the scope of this study.

5.2 Research Questions

The aim of this chapter is to conduct an end-to-end evaluation of the proposed framework and dataset to determine their effectiveness, complementarity, and potential for real-world integration. In this context, end-to-end refers to evaluating the complete research pipeline, to post-hoc expert assessment of both the framework and the dataset. Specifically, this chapter seeks to understand how these two components work together to inform the design of future technostress-aware development environments.

(Obj 1) To evaluate the perceived usefulness, novelty, and applicability of the proposed technostress-aware framework among MSc students and Developers.

(Obj 2) To assess the credibility, research utility, and reusability of the TIDE dataset as a supporting resource for affective and technostress-aware computing.

RQ1 (Obj 1) examines how developers and MSc User Experience Design students perceive the proposed technostress-aware IDE enhancement framework. This ensures that the framework is evaluated. This ensures that the framework is not only evaluation for the outcomes of the modifications it suggests, but also for its usage by its primary users - the designers and developers . RQ2 (Obj 2) focuses on how the same stakeholders evaluate the TIDE dataset as a complementary empirical resource to the framework. This research question assesses whether the dataset is viewed as a meaningful and trustworthy evidence base for supporting future affective and technostress-aware investigations within similar IDE-mediated contexts.

To maintain conceptual consistency with the broader evaluation criteria, these objective-specific constructs are aligned with the three overarching assessment dimensions. Perceived usefulness and credibility reflect potential for real-world integration, as they indicate whether the framework and dataset are considered viable and trustworthy in practice. Applicability and reusability correspond to effectiveness, capturing the extent to which the contributions can function as intended. Novelty and research utility align with complementarity, evaluating how the framework and dataset extend, enhance, or integrate with existing approaches.

To achieve these objectives, the study addresses the following research questions:

(RQ 1) How do developers and MSc User Experience Design students perceive the usefulness, novelty, and applicability of the proposed technostress-aware IDE enhancement framework?

(RQ 2) How do developers and MSc User Experience Design students evaluate the research utility, credibility, and future potential of the TIDE dataset as a complementary resource to the framework?

5.3 Setup

5.3.1 Procedure

The evaluation procedure (Figure 5.1) comprised several structured stages designed to rigorously assess both the proposed framework and the biometric dataset from the perspectives of MSc students and professional developers. The multi-step evaluation structure ensured that the analysis was systematic, balanced, and reflective of the needs and experiences of distinct user communities.

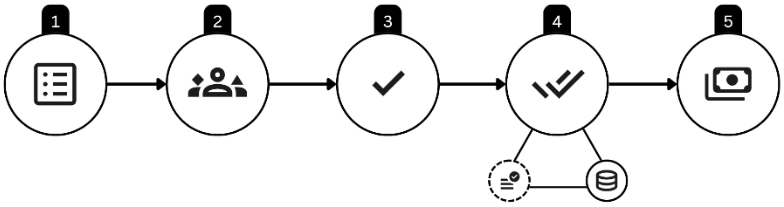


Figure 5.1: Evaluation Procedure

5.3.1.1 Survey Design

The evaluation instruments used in this study were developed specifically to assess perceptions of the proposed framework and dataset. Questionnaire items were informed by the research objectives, findings from the preceding chapters, and established evaluation constructs commonly used within technology acceptance and software-engineering research. Prior to deployment, the instruments were reviewed by the supervisory team to ensure clarity, relevance, and alignment with the study objectives. Feedback from this review process was incorporated through iterative refinement of question wording and structure.

The questionnaires should be considered researcher-developed instruments rather than formally validated measurement scales. While efforts were made to improve face validity through supervisory review and iterative refinement, comprehensive psychometric validation, including exploratory factor analysis, confirmatory factor analysis, or external validation across independent samples, was beyond the scope of this research. Therefore, findings derived from these instruments should be interpreted as exploratory evaluations of participant perceptions rather than definitive measures of framework or dataset acceptance.

5.3.1.1.1 Framework

The six framework evaluation factors (Table. 5.1) (F1–F6) were conceptually aligned with three higher-level evaluative dimensions: usefulness, novelty, and applicability. The dimension of usefulness encompassed Clarity (F1), Effectiveness (F2), and Technostress Reduction (F5), as these factors collectively assess how well the framework functions in practice, its ease of understanding, and its capacity to deliver meaningful benefits in reducing technostress within IDEs. Novelty was primarily represented through Integration (F4), which reflects the framework's originality and its potential to serve as a reference model for future research and IDE development. Finally, applicability was captured by Practicality (F3) and Confidence (F6), which examine the framework's relevance to real-world software design contexts and the likelihood of its adoption by developers or researchers. Together, these mappings provide a structured lens for interpreting participant evaluations across the key dimensions of the framework's value and contribution. The framework section (which addresses Obj 1) of the survey is subsequently followed by the dataset section (which addresses Obj 2), creating a natural progression between the two contributions.

Table 5.1: Framework Section of Survey

Framework		1	2	3	4	5
F1*	*Clarity*: The process of collecting and analysing feedback within this framework is clearly explained and easy to follow.					
F2*	*Effectiveness*: The structured approach (i.e., user feedback + data-driven refinement) is effective in improving developer experience within IDEs.					
F3*	*Practicality*: A feedback-driven, technostress-aware framework is relevant for modern IDE design.					
F4*	*Integration*: This framework could serve as a model for future IDE research and development.					
F5*	*Technostress Reduction*: This framework has the potential to reduce technostress for developers using IDEs.					
F6*	*Confidence*: I would be likely to adopt this framework if I were to develop mental well-being-aware IDEs.					
Scale: 1-Strongly Disagree, 2-Disagree, 3-Neutral, 4-Agree, 5-Strongly Agree						

In addition, two open ended questions were also asked regarding the framework (Table. 5.2)

Table 5.2: Framework Section of Survey Continued

F7*	In your own words, what are your thoughts on the framework and the survey dataset?
F8	In your own words, what improvements or additional features would you suggest to make this feedback-driven, technostress-aware framework more effective for IDE development?

5.3.1.1.2 Dataset

The seven evaluation factors (Table. 5.3) (D1–D7) of the TIDE biometric dataset were conceptually aligned with three overarching evaluative dimensions: credibility, research utility, and reusability. The dimension of credibility encompasses Clarity (D1) and Confidence (D6), as these factors assess the dataset's transparency, documentation quality, and trustworthiness in representing authentic user emotions and interactions. Research utility is reflected through Effectiveness (D2), Integration (D4), and Technostress Reduction (D5), which collectively evaluate the dataset's analytical value, its potential to enhance the accuracy of IDE improvements, and its capacity to inform interventions aimed at mitigating technostress. Finally, reusability is captured by Practicality (D3) and Emotional Markers (D7), which consider the dataset's applicability for future similar studies, its potential to fine-tune technostress-aware IDE features, and the representativeness of the emotional markers selected for analysis. Together, these mappings position the TIDE dataset as a preliminary expert assessed, research-driven, and reusable resource within comparable experimental and IDE contexts, capable of supporting ongoing work in technostress detection, affective computing, and adaptive software design, subject to the scope and limitations discussed under threats to validity in Chapter 4.

Table 5.3: Dataset Section of Survey

Dataset		1	2	3	4	5
D1*	*Clarity*: The structure and contents of the biometric dataset are clearly explained and easy to understand.					
D2*	*Effectiveness*: The inclusion of facial emotion, and phase-based segmentation is useful for studying developer experience.					
D3*	*Practicality*: The biometric dataset can be effectively used to fine-tune technostress-aware IDE features in future versions.					
D4*	*Integration*: Use of biometric data would enhance the overall accuracy and relevance of IDE improvements.					
D5*	*Technostress Reduction*: Insights from biometric data have the potential to reduce technostress in IDE users.					
D6*	*Confidence*: The biometric dataset is relevant for designing user-friendly software using representative emotional markers.					
D7*	*Emotional Markers*: Joy, Surprise, Engagement, Confusion, Anger, Sadness, and Attention are representative of the emotional markers you associate with IDE interaction.					
Scale: 1-Strongly Disagree, 2-Disagree, 3-Neutral, 4-Agree, 5-Strongly Agree						

Once again two addition, two open ended questions were also asked regarding the biometric dataset (Table. 5.4)

Table 5.4: Dataset Section of Survey Continued

D8*	In your own words, what are your thoughts on the using biometric data as a source of additional insight to qualitative/quantitative data from surveys in the previous question?
D9	Any further comments?

5.3.1.2 Distribution

The survey was distributed to two participant groups: MSc students and professional developers. MSc participants were recruited through academic channels, with course leads sharing the study details via institutional email and in-class announcements. Developer participants were reached through professional networks, including mailing lists and posts shared on LinkedIn, both on personal profiles and within relevant professional groups. In this study, a “developer” was defined as an individual whose professional responsibilities involve writing, modifying, integrating, or maintaining code as a core part of their role. This definition aligns with the actual participant pool, which included individuals working as Data Engineers, Software Engineers, Software Integration Engineers, DevOps Engineers, and SAP Application Developers. These roles require routine engagement with programming, debugging, and working with development environments, ensuring that all participants possessed the practical expertise necessary to meaningfully evaluate the IDE modifications assessed in this study.

5.3.1.3 Participant Verification

Verification was undertaken to ensure that respondents belonged to the intended target demographics, namely MSc computing students and professional developers. This step was necessary because, when surveys are distributed via open or semi-open channels, there is a risk that they may be shared beyond the intended audience (e.g., through peer forwarding), leading to responses from individuals outside the study scope. Ensuring demographic validity therefore helped maintain the integrity and relevance of the collected data. Upon receiving the survey submissions, a structured verification and quality assurance process was implemented to ensure data reliability and participant authenticity. For MSc participants, verification required the provision of a valid university ID, email address, and course name. Developer participants were required to provide their professional position, company name, and work email to confirm eligibility.

5.3.1.4 Data Verification

Following participant verification, all submissions underwent an internal review conducted by the primary researcher to assess data completeness and consistency between qualitative and quantitative responses. Any entries identified as anomalies were referred to a member of the supervisory team for secondary evaluation, ensuring methodological rigor and maintaining data integrity.

5.3.1.5 Reward

Once the review process was complete, participants whose submissions were successfully validated were formally notified via email. Eligible participants received a digital voucher as compensation for their time and contribution to the study.

5.4 Sociodemographic

As a result of following the aforementioned procedure, a total of 18 participants initially opted to take part in the evaluation. Of these, data from 17 participants were deemed valid for analysis (Table. 5.5). One MSc participant was excluded because their quantitative responses were inconsistent with their qualitative feedback, suggesting potential issues with data reliability. Among the valid participants, seven were professional developers and ten were Postgraduate MSc students.

The demographic composition of the sample is reported for contextual and descriptive purposes only. Given the relatively modest sample size, the participant characteristics are not intended to support statistical comparisons between groups or broader generalisations regarding developer or student populations.

Table 5.5: Participant Demographic for Evaluation

P	***Gender***	***Age***	***Status***	***Experience***	***Ethnicity***
1	PNtS	Over 25	Developer	Over 2	White
2	Male	Over 25	Developer	Over 2	White
3	Male	Over 25	Developer	Over 2	Black
4	Male	18 - 25	MSc Student	1 - 2	Asian
5	Male	18 - 25	MSc Student	Over 2	Asian
7	Male	18 - 25	MSc Student	1 - 2	Black
8	Male	Over 25	Developer	1 - 2	Asian
9	Male	18 - 25	MSc Student	Under 1	Asian
10	Male	Over 25	Developer	Over 2	Asian
11	Female	18 - 25	MSc Student	Under 1	Middle East
12	Male	18 - 25	MSc Student	Under 1	Asian
13	Male	18 - 25	Developer	Over 2	Black
14	Male	18 - 25	MSc Student	Under 1	Asian
15	Female	Over 25	MSc Student	Over 2	Mexican
16	Male	Over 25	MSc Student	1 - 2	Black
17	Female	Over 25	Developer	Over 2	Asian
18	Female	Over 25	MSc Student	1 - 2	Black

When comparing gender distribution between the Developer and MSc groups (Table. 5.6), both showed a relatively similar male-to-female ratio, indicating a balanced representation across both cohorts. In terms of age, the MSc participants were predominantly within the 18–25 age group (70%), reflecting their status as early-career or postgraduate students. Conversely, the majority of developers (around 86%) were aged over 25, which aligns with their greater industry experience and professional background.

Ethnic diversity was evident across both groups, with participants representing a broad range of ethnic backgrounds. This variation adds valuable perspective to the evaluation, ensuring that the findings are not limited to a single cultural or demographic context. Regarding disability representation, only one MSc participant identified as having a disability, suggesting that future research could aim to achieve broader inclusion in this area. While the later analysis does not explicitly dissect findings by these demographic categories, this information remains important for establishing the representativeness and reliability of the data. The demographic spread observed among the MSc and developer participants was largely consistent with that of the undergraduate cohort from the earlier experiment (Chapter Three and Four), particularly in terms of gender balance, age range, and ethnic diversity. This alignment suggests a degree of continuity across samples, which helps reduce the risk of demographic bias influencing the feedback or the interpretation of the dataset. Furthermore, including demographic data supports transparency and contextual understanding of the participants' backgrounds. It provides reassurance that the insights gathered from the evaluation are not limited to a narrow demographic but instead reflect perspectives across a range of experiences and identities. This consistency also strengthens the validity of the comparison between the experiment and the framework evaluation, ensuring that differences in feedback or outcomes are more likely to be attributed to the framework or IDE modifications rather than demographic variability.

In terms of programming experience, most MSc students reported having either less than one year (40%) or between one and two years (40%) of experience, reflecting their ongoing development of technical skills. In contrast, the majority of developers (86%) reported more than two years of programming experience, demonstrating their higher level of expertise and professional familiarity with development environments. While some undergraduate students in the experiment (Chapter 3) showed comparable years of programming experience, this group is distinct in both context and purpose. The MSc User Experience Design students are not currently studying programming as their primary discipline but are instead engaging with it from a design and usability perspective, focusing on how technology impacts user interaction and technostress. Conversely, the developers possess industry-based experience, making them the primary intended users of the framework and dataset, those capable of integrating technostress-aware principles into IDEs or other software tools. Together, these two groups represent a more advanced and professionally aligned perspective, evaluating the framework not as learners but as potential designers and implementers of future technostress-integrated development environments.

Table 5.6: Comparison of Participant Demographic for Evaluation

	Overall Count	***Overall %***	***Dev Count***	***Dev %***	***MSc Count***	***MSc %***
Gender						
Male	12	71%	5	71%	7	70%
Female	4	24%	1	14%	3	30%
PNtS	1	6%	1	14%	0	0%
Age						
Under 18	0	0%	0	0%	0	0%
18 - 25	8	47%	1	14%	7	70%
Over 25	9	53%	6	86%	3	30%
Ethnicity						
White	2	12%	2	29%	0	0%
Black	5	29%	2	29%	3	30%
Asian	8	47%	3	43%	5	50%
Middle East	1	6%	0	0%	1	10%
Mexican	1	6%	0	0%	1	10%
Experience						
Under 1	4	24%	0	0%	4	40%
1 - 2	5	29%	1	14%	4	40%
Over 2	8	47%	6	86%	2	20%

5.5 Framework Evaluation

Although sociodemographic information such as ethnicity, age, experience level, and nationality/region is reported to contextualise the participant sample, the evaluation in this chapter does not disaggregate quantitative findings by these variables. Unlike Chapter 3, where sociodemographic comparisons formed part of the analytical scope, the research questions in this chapter focus on evaluation at a collective level rather than across subgroup differences. Consequently, demographic data are presented to ensure transparency and sample characterisation, but they are not treated as analytical variables within this phase of the study.

To maintain analytical consistency, the interpretation of Likert-scale responses in this section adheres to the same categorisation thresholds and evaluative criteria defined in Chapter 3 (Section 3.4.4.3.1.6). This ensures that comparisons across study components are methodologically aligned and conceptually coherent.

5.5.1 Quantitative Analysis

The quantitative evaluation results (Figure. 5.2) indicate positive perceptions across all six framework criteria (F1 to F6) from both developers and MSc Computing participants.

Clarity (**F1**) received an overall mean of 4.06 (Agree), with developers (M = 4.43, SD = 0.53) rating it higher than MSc students (M = 3.80, SD = 0.63). This suggests that developers found the framework's feedback collection and analysis processes easier to understand, possibly due to their greater familiarity with iterative development and evaluation workflows. The SEM intervals for developers (4.23–4.63) and MSc participants (3.60–4.00) do not overlap, as the developer lower bound exceeds the MSc upper bound. This non-overlap suggests a potentially statistically meaningful difference in the precision-adjusted estimates between the two cohorts.

Effectiveness (**F2**) and Practicality (**F3**) also achieved overall ratings of 4.06 (Agree), indicating that both participant groups recognised the framework's structured, data-driven approach as an effective method for improving developer experience within IDEs. The similarity between developer and MSc ratings suggests shared confidence in its relevance to both professional and academic settings. For F2 the SEM ranges for developers (3.74–4.54) and MSc (3.70–4.30) overlap substantially, indicating that the difference between cohort estimates cannot be considered statistically distinct based on SEM. Similarly, for F3 the SEM intervals for developers (3.62–4.38) and MSc (3.87–4.33) also overlap, suggesting no meaningful separation between groups.

Integration (**F4**) achieved the highest overall mean (4.24, Strongly Agree), with developers (M = 4.29) and MSc participants (M = 4.20) strongly agreeing that the framework could serve as a foundational model for future IDE research and development. This reflects strong endorsement of the framework's theoretical robustness and its potential as a replicable tool for guiding future work. Strong overlap is observed between the Developer (4.11–4.47) and MSc (4.07–4.33) SEM ranges, indicating consistent estimates across cohorts.

Technostress Reduction (**F5**) received an overall mean of 4.12 (Agree), again rated slightly higher by developers (M = 4.29) than MSc participants (M = 4.00). This difference suggests that practitioners, who are more accustomed to prolonged IDE use were more confident in the framework's ability to alleviate technostress through adaptive, user-centred refinement strategies. For F5, although the developer estimate is higher (4.11–4.47), the SEM intervals overlap with MSc (3.79–4.21), suggesting that the difference is not clearly statistically distinct.

Finally, Confidence and Adoption Intent (**F6**) showed the lowest overall score (3.71, Agree), with both groups providing near-identical ratings (Dev_M = 3.71, MSc_M = 3.70). While still positive, this suggests cautious optimism regarding direct adoption, implying that further empirical validation and integration into existing workflows may enhance its perceived usability and real-world applicability. F6 shows wide and overlapping SEM ranges (Developer: 3.29–4.13; MSc: 3.33–4.07), indicating once again no meaningful difference in precision-adjusted estimates between the two groups.

Collectively, the results indicate strong cross-cohort agreement across most framework dimensions. Visual comparison of the mean scores and associated SEM intervals suggests broadly similar perceptions between developers and MSc participants across the majority of constructs. Clarity (F1) exhibited the largest observed difference between the two groups; however, this comparison is descriptive in nature and should not be interpreted as evidence of statistical significance. Overall, the findings reinforce the framework's novel contribution to IDE enhancement research and its potential to inform the development of technostress-aware, feedback-driven programming environments aimed at improving both user wellbeing and experience.

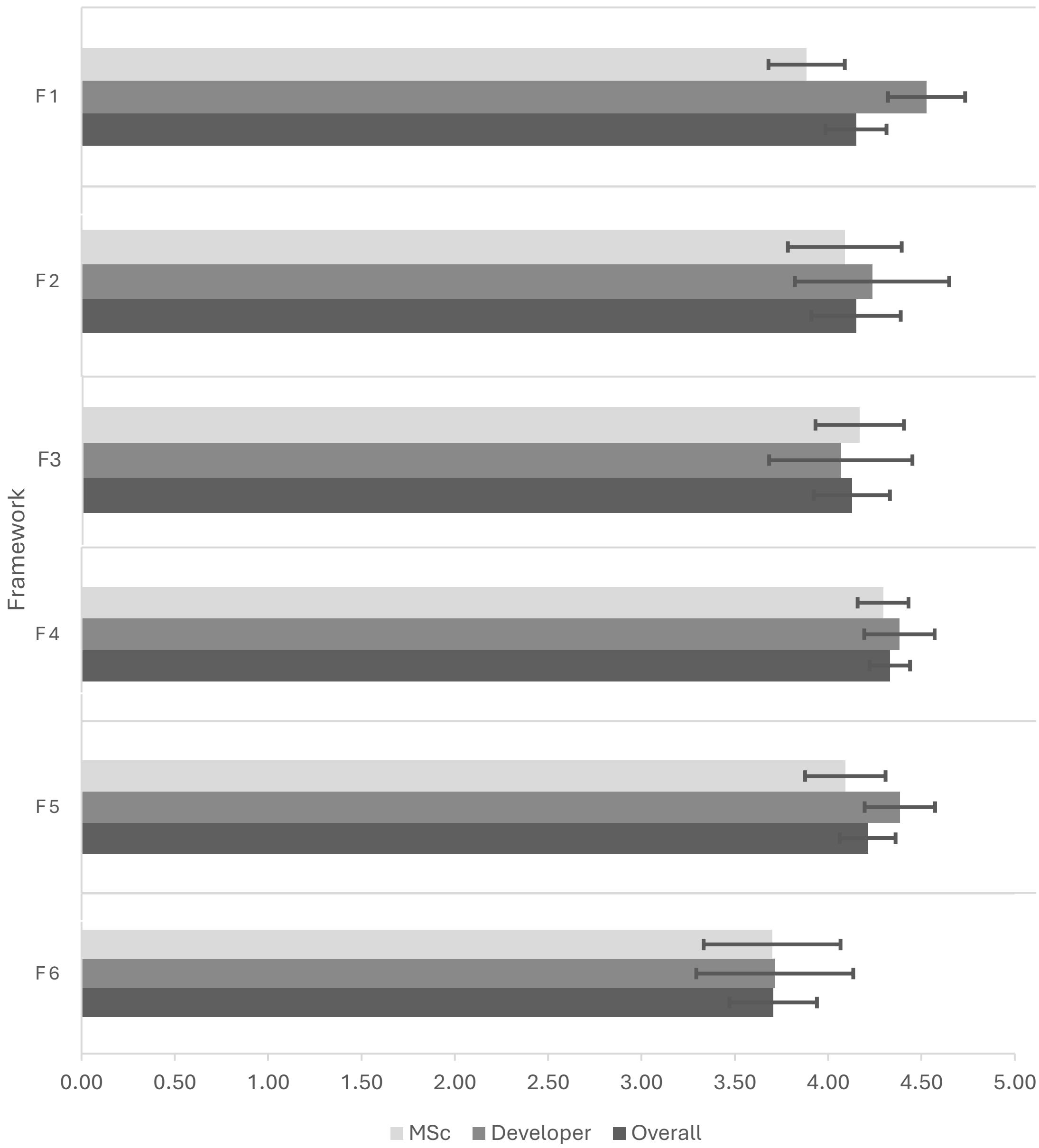


Figure 5.2: Framework Evaluation Scores

5.5.1.1 Internal Consistency

The internal consistency of the six-item framework evaluation scale (Clarity, Effectiveness, Practicality, Integration, Technostress Reduction, and Confidence) was assessed using Cronbach's Alpha, yielding a coefficient of $\alpha = 0.57$, which indicates moderate reliability. While this value is below the commonly accepted threshold of 0.70 for confirmatory studies, it remains acceptable in exploratory research, particularly when assessing complex and multidimensional constructs [377]. Consequently, the aggregated framework evaluation score should be interpreted with appropriate caution.

This outcome can be partly attributed to the conceptual breadth of the framework evaluation questions. Each item targets a distinct yet interrelated aspect of perceived framework quality, from clarity of process to perceived adoption confidence, rather than a single unified construct. Such diversity in item content can naturally lower inter-item correlations and, consequently, the alpha coefficient. In this case, the instrument captures multiple evaluative perspectives, including both cognitive (clarity, understanding) and behavioural (confidence, likelihood of adoption) components, reflecting the framework's multi-dimensional nature.

However, this result should be considered in the context of the instrument's intended purpose. The framework evaluation was designed to assess multiple dimensions of framework perception rather than a single underlying construct. As Cronbach's Alpha assumes that items measure a common latent construct, lower alpha values are often observed when instruments intentionally assess multiple related but conceptually distinct dimensions. Therefore, a modest alpha coefficient does not necessarily indicate poor item quality but may instead reflect the multidimensional nature of the evaluation instrument.

Furthermore, the relatively small sample size ($n = 17$) may have contributed to instability in the reliability estimate. Cronbach's Alpha is sensitive to sample size, and small variations in participant responses can have a proportionally greater effect on the resulting coefficient. The mixed participant composition, consisting of both MSc students and professional developers, may also have introduced additional response variability, as participants may have evaluated the framework from different perspectives and usage contexts [378].

Given these considerations, the framework evaluation findings are best interpreted at both the aggregate and item levels. Individual dimensions provide valuable insight into specific strengths and weaknesses of the framework, while the overall score should be viewed as a broad exploratory indicator of participant perceptions rather than a definitive measure of a single construct

5.5.2 Qualitative Analysis

The qualitative responses relating to the proposed framework were analysed using descriptive lexical visualisation techniques. Specifically, a word cloud was generated to highlight the most frequently occurring terms within participant feedback concerning the framework. The purpose of this analysis was to provide an accessible overview of the language participants used when describing their perceptions of the framework. The resulting visualisation should not be interpreted as a formal qualitative analysis method, such as thematic analysis, nor should word frequency be considered a direct indicator of importance. Instead, the word cloud serves as descriptive lexical evidence that complements the quantitative evaluation results by illustrating recurring concepts and participant emphasis areas.

5.5.2.1 Lexical Divergence

The lexical divergence between cohorts reflects a fundamental difference in how the methodological framework is conceptualised. Developers (Figure. 5.3) predominantly treat the framework as a structured analytical artefact to be implemented, organised, and operationalised. Their discourse emphasises its novelty, its relevance to developers, its integration of technostress, and its structured analytical design, indicating an orientation toward architectural coherence and methodological construction. The framework is framed as a formal model that integrates constructs in a systematic manner, with attention directed toward clarity of structure, logical organisation, and applicability.

Figure 5.3: Developer Framework Word Cloud

In contrast, MSc participants (Figure. 5.4) treat the framework as an interpretive mechanism intended to generate meaningful insight and enhance understanding of technostress phenomena. While the framework remains central in their discourse, the emphasis shifts toward data, insight generation, clarity, relevance, and practical use. This suggests that the framework is evaluated not solely as a structural model, but as a conceptual lens through which complex empirical relationships can be interpreted. Rather than focusing primarily on architectural organisation, this cohort foregrounds explanatory utility and conceptual clarity.

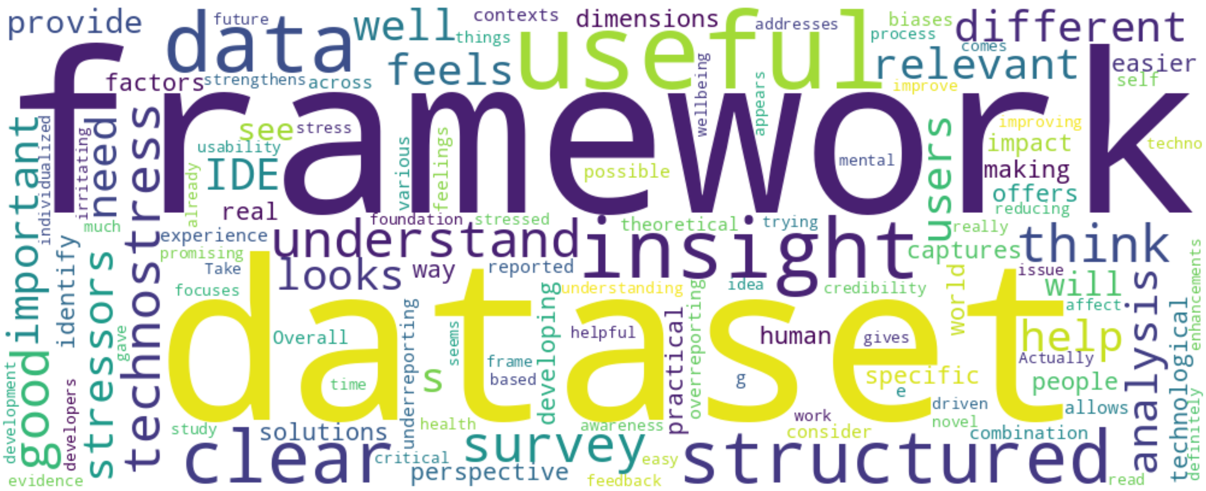


Figure 5.4: MSc Framework Word Cloud

The distinction is epistemic rather than topical. Both cohorts recognise the framework as an integrative model linking technostress and user experience constructs; however, developers approach it from a systems-oriented and implementation perspective, whereas MSc participants approach it from an interpretive and insight-oriented perspective. This difference again reflects expertise-dependent cognitive framing, with developers privileging structural coherence and operational feasibility, and MSc students privileging explanatory value and conceptual usefulness.

5.5.2.2 Sample Responses

5.5.2.2.1 Developer

Overall, the responses (for **F7**) indicated strong agreement on the framework's structural clarity and logical organisation. Several participants described it as intuitive and easy to follow.

> *P3 noted that "the framework is well organised and makes it easy to understand what is being measured. The dataset too is simple to understand. The variable/column names are easy to make analysis on since they are straight forward."*
>
> *Likewise, P13 observed that "the framework is well-structured, with each stage building logically on the last. I like how it combines both qualitative and quantitative measures, making the dataset more reliable and reflective of real user experiences. The surveys are thorough, and by validating participants' skills, they strengthen the credibility of the findings."*
>
> *P8 similarly described it as "well structured and effective."*

These responses highlight that participants viewed the framework as methodologically robust and comprehensible, with the inclusion of both qualitative and quantitative measures enhancing its reliability and depth. The emphasis on participant validation, as highlighted by P13, further reinforces the framework's credibility and transparency, crucial aspects when assessing human–technology interaction within complex learning or work environments.

Beyond structure, participants also recognised the framework's novelty and relevance to real-world developer experiences.

> *P10 remarked that "I wasn't previously aware of similar frameworks, so the concept stands out as quite novel. Addressing technostress directly inside developer tools has real potential value it seems. Even if we don't always call it that, technostress is something most developers run into. It's not just about productivity, it's also about making the day-to-day easier."*

This response underscores the framework's practical orientation, demonstrating how its integration within IDEs could serve as both a research mechanism and an applied intervention for improving daily workflows and mitigating technostress.

> *Similarly, P17 expanded on its potential relevance to industry, stating that "it is a great idea to introduce these frameworks and emphasis the importance of these frameworks in the corporate world where as developers we tend to spend most of our hours developing and coding in different platforms, and having such frameworks installed in the actual application and our daily workflows would not only raise awareness about technostress but also highlight the importance of mental well being in tech driven environments."*

This perspective illustrates the perceived scalability of the framework, suggesting it could extend beyond academia into workplace ecosystems, promoting both productivity and mental wellbeing. In addition, some developers provided feedback that reflected anticipation and curiosity about the framework's practical application.

> *P1 expressed that it "seems interesting, considering implementation with terminal as well," highlighting opportunities for expanding its scope across multiple IDE components.*
>
> *P2 similarly noted it was a "promising statement" and expressed being "intrigued to see the results,"*

Indicating interest in observing how the framework performs when applied in practice. Such remarks highlight a general sense of optimism among participants, suggesting that the framework is not only conceptually sound but also practically compelling.

The qualitative feedback from developers indicates a strong agreement regarding the framework's structure, clarity, and real-world applicability. Participants viewed it as a novel and well-organised approach that effectively integrates both quantitative and qualitative dimensions, thereby enhancing data reliability. Moreover, its focus on embedding technostress awareness within everyday development environments was seen as both timely and valuable.

When asked about possible improvements or additional features to enhance the framework's effectiveness for IDE development (for **F8**), participants provided a range of insightful and practical suggestions.

> *P3 proposed that "real-time stress indicators as well as more granular feedback options" would improve the framework's usefulness, explaining that "the former makes the framework easy for developers to see when they are getting overwhelmed and the latter helps them pinpoint the cause of the stress."*

This highlights the importance of immediate and specific technostress detection, suggesting that static or post-task feedback may not be sufficient to capture the dynamic nature of stress during coding activities.

> *Similarly, P13 recommended integrating "real-time feedback options and lightweight stress indicators, such as keystroke patterns, to complement surveys." They further suggested that "adaptive tasks could capture technostress across different skill levels, while a visualisation dashboard would make patterns and improvements clearer over multiple cycles."*

These ideas align closely with the iterative principles of the framework itself, which already incorporates cyclical evaluation and refinement stages. Through its feedback loop, the framework allows the continuous collection and analysis of user responses, enabling patterns to be identified and adjustments to be made over multiple iterations, directly addressing the need for adaptive improvement and visual insight raised by participants.

> *Personalisation also emerged as a key area of improvement. P10 emphasised that "when we face technostress, we don't want to add it, and that doesn't add to it." They also noted that "personalisation's for each developers might be a good improvement, especially when working with many different developers [who] have different habits, such as getting distracted or overloaded."*

This underscores the importance of ensuring that any stress monitoring or feedback mechanism remains lightweight and non-intrusive, while adapting to individual behavioural patterns. The framework's iterative cycle helps to mitigate this concern by refining both data capture and feedback presentation methods based on user input, ensuring that future iterations become more tailored and less cognitively demanding over time.

> *P17 offered a particularly holistic perspective, arguing that for the framework to "support developers in a meaningful way, it needs to be able to adapt to how people are actually feeling and working over time." They elaborated that this could include "mood tracking, personalised reminders to take breaks, and even simplified UI modes when someone's clearly under pressure." Importantly, P17 extended the scope beyond individual experience to team culture, proposing that "sharing anonymised insights like general stress trends or break habits could help teams be more empathetic without making anyone feel singled out."*

Collectively, these views reveal strong interest in real-time monitoring, personalisation, and long-term adaptability. The existing iterative feedback loop embedded in the framework already provides a foundation for addressing many of these needs, by enabling continuous refinement based on user data and experiential feedback. Over time, such an approach can evolve to include additional capabilities such as adaptive user interfaces, dynamic stress visualisation, and aggregated trend analysis to support both individual wellbeing and team-level awareness within IDE environments.

5.5.2.2.2 Postgraduate (MSc)

When asked for their thoughts on the framework and the survey dataset (for **F7**), MSc participants generally shared positive and thoughtful feedback, often recognising the framework's clarity, structure, and potential impact.

> *P4 shared that "the framework provides a clear and structured way to understand the different dimensions of technostress, making it easier to identify specific stressors and their impact on users." They appreciated that it "captures both the technological and human factors, which is important for developing practical solutions." Regarding the dataset, they noted that it "offers real-world insight into how people experience technostress across various contexts," while also acknowledging that "as with any self-reported data, it's important to consider possible biases." Overall, concluding that "the combination of the framework and dataset allows for both theoretical understanding and evidence-based analysis, which strengthens the study's credibility."*

Several participants also commented on the framework's clear design and the quality of the dataset.

> *P5 mentioned that it was "well structured and the process was clear" and that "the data gave me good insights."*
>
> *Similarly, P14 stated that "the framework is clear and easy to understand and the dataset feels relevant and gives useful insights."*
>
> *P7 expressed optimism about the research direction, stating that "the framework looks promising because it addresses a critical issue and the survey dataset appears to have a good foundation for data analysis."*

Some participants highlighted how the framework could be beneficial in practical applications.

> *P11 commented that "the framework seems very useful and relevant because it focuses on reducing technostress while improving IDE usability." They added that "the idea of feedback driven enhancements feels novel and could really help developers in the future," and that "the survey dataset also looks very well structured and should provide good insights into how different techno stressors affect users."*

Other MSc participants provided shorter but meaningful feedback.

> *P12 stated simply that "I think it will be helpful,"*
>
> *P16 emphasised the wellbeing aspect, saying "I think the framework will be useful for mental health wellbeing awareness."*
>
> *Finally, P15 reflected on the need for more personalisation, explaining that "I can definitely see where it could help with the stress that comes with IDE development but I think it would need to be more individualized and could see where it trying to improve things could be irritating when you're already stressed."*

On the whole, MSc participants viewed the framework as clear, useful, and relevant, especially in addressing technostress and supporting mental wellbeing. Their feedback also pointed to opportunities for refinement, such as personalisation and bias awareness, which can be addressed through the framework's iterative feedback approach.

When asked about potential improvements features to make the feedback-driven, technostress-aware framework more effective for IDE development (for **F8**), MSc participants focused on ideas that emphasised personalisation, real-time interaction, and adaptive support.

> *P4 suggested adding "a simple real-time stress indicator in the IDE (like a gentle alert or colour change), offering quick tips or breaks when stress signals are detected." They also proposed that "including customisable settings so developers can choose which stress factors to monitor would make it more personal and effective." This feedback highlights a strong desire for the framework to provide both real-time feedback and individual control,*

Aligning well with the framework's iterative feedback loop, which is designed to adapt to user responses over time.

> *Similarly, P7 stated that "perhaps some sort of personalisation or customisation that allows developers to customise based on their personal preferences" would improve usability. P9 also focused on the importance of engagement and comfort, suggesting that "I'd design the framework to be more interesting for the first sight, tailored more to each person, and embedded it into the IDE, so the feedback helps developers stay focused and comfortable instead of adding stress."*

Several students mentioned adding adaptive and visual features to make the framework more supportive.

> *P14 proposed "adding reminders and adaptive features, and simple visuals could make it more effective.", P18 recommended expanding the framework's evaluation metrics by saying "add a metric to rate the scale of all groups when completing programming tasks, so that it's relevant to participants in each group. Some type of tracker to give real-time feedback."*

P15 discussed the potential role of advanced AI within the system,

> *Noting that "LLMs are a great addition but if it could be calibrated to a user before starting that would be helpful."*

MSc participants proposed a more dynamic, personalised, and interactive version of the framework. Their suggestions, including real-time stress indicators, adaptive reminders, and individual calibration, align with the framework's feedback-driven structure, which can evolve based on user input and biometric data. Integrating these enhancements would strengthen the system's ability to proactively support developers' wellbeing while maintaining usability and relevance across different experience levels.

5.5.3 Addressing Research Question One

Question: How do developers and MSc User Experience Design students perceive the usefulness, novelty, and applicability of the proposed technostress-aware IDE enhancement framework? (RQ1)

Findings: Both developers and MSc User Experience Design students evaluated the framework positively, describing it as a well-structured and practical foundation for understanding and addressing technostress in programming environments. The qualitative evaluation revealed that participants from both groups appreciated the framework's systematic design, which revolves around collecting user feedback, analysing it through measurable indicators, and applying targeted modifications. This structured approach was viewed as a major strength, effectively bridging theoretical understanding with real-world application in the context of IDE development.

While both groups recognised the framework's clarity and relevance, developers generally expressed stronger confidence in its immediate practicality and potential for real-world implementation. Their familiarity with iterative workflows and software evaluation processes made them more attuned to how the framework could be integrated within professional development environments. They particularly valued its data driven approach, which aligns with established software refinement practices and offers a clear pathway for enhancing user experience and productivity.

In contrast, MSc User Experience Design students approached the framework through a design and user-centred lens. They appreciated its structured methodology and recognised its potential to guide future research but emphasised the importance of personalisation, real-time feedback, and adaptability. Their feedback indicated that while the framework already provides a strong foundation for identifying and mitigating technostress, it could evolve beyond its current stage, moving from a developer mediated model toward a more autonomous and adaptive system capable of making real-time improvements.

Participants from both groups envisioned a natural progression for the framework, transforming it into a live, self-improving feedback loop embedded directly within the IDE. In this envisioned future state, the IDE would continuously gather user input such as interaction patterns, biometric cues, and self-reported sentiment, and automatically adjust interface elements to optimise user comfort, focus, and wellbeing. Such an adaptive process would make IDEs more responsive to individual needs and position the framework as a foundation for developing technostress-aware software systems that actively support mental health and performance.

Discussion: Both developers and MSc participants perceived the proposed framework as a useful, novel, and applicable approach to improving user experience and reducing technostress in IDEs. The framework's structured design and integration potential were recognised as key strengths, while its future development into a live, adaptive system was seen as a natural progression. Collectively, these findings provide preliminary expert assessment to the framework's relevance and support its value as a foundation for creating technostress-aware, feedback-driven IDEs.

5.6 Dataset Evaluation

5.6.1 Quantitative Analysis

The quantitative evaluation of the TIDE biometric dataset (Figure. 5.5) revealed an overall positive perception from both developers and MSc Computing participants. The results indicate that the dataset is viewed as clear, relevant, and practically valuable for advancing research into wellbeing and technostress-aware IDEs. Overall mean ratings were consistently above the neutral midpoint, suggesting strong agreement across participants regarding its structure, utility, and research potential.

Clarity (**D1**) received an overall mean of 3.94 (Agree), with developers (M = 4.00, SD = 1.00) and MSc students (M = 3.90, SD = 0.88) expressing consistent views that the dataset's structure and content were clearly described and easy to interpret. This indicates that both groups found the documentation and representation of emotion metrics, sufficiently transparent. The SEM intervals for Developer (3.62–4.38) and MSc (3.62–4.18) overlap substantially, indicating no statistically distinct difference between the two cohorts.

Effectiveness (**D2**) achieved one of the strongest mean scores (O_M = 4.12, Agree), reflecting that participants found the inclusion of facial emotion, and phase-based segmentation useful for studying developer experience. Developers rated this higher (M = 4.29) than MSc participants (M = 4.00), suggesting that those with greater technical exposure recognised its analytical value in understanding emotional and cognitive dimensions of programming. For D2, the SEM ranges Developer (3.87–4.71) and MSc (3.67–4.33) also overlap, suggesting that the difference in estimates is not clearly statistically meaningful.

Practicality (**D3**) also scored highly (O_M = 4.06, Agree), indicating that both groups viewed the dataset as feasible for fine-tuning future technostress-aware IDE features. The consistency between developer (M = 4.00) and MSc (M = 4.10) responses highlights shared confidence in its applicability for real-world and research-based use. In D3, the SEM intervals Developer (3.62–4.38) and MSc (4.00–4.20) overlap, despite the higher precision observed in the MSc group. This indicates no meaningful separation between cohorts.

Integration (**D4**) showed a moderate positive response (O_M = 3.82, Agree), suggesting that participants generally agreed that biometric data could enhance the accuracy and relevance of IDE improvements. Once again Developer (3.52–4.20) and MSc (3.51–4.09) SEM ranges overlap considerably, suggesting comparable precision-adjusted estimates.

Technostress Reduction (**D5**) and Confidence (**D6**) both achieved mean scores of 3.71 (Agree). Developers rated these items higher (M = 4.00 and M = 4.29 respectively) compared to MSc participants (M = 3.50 and M = 3.30), showing that industry professionals were more confident in the dataset's ability to inform technostress-aware IDE design. This difference may stem from practitioners' greater understanding of how biometric indicators can be applied in real-time systems to mitigate stress and cognitive overload. In D5, the SEM intervals Developer (3.78–4.22) and MSc (3.13–3.87) show a slight boundary overlap (3.78–3.87). Although the overlap is minimal, it prevents a clear conclusion of statistical separation. However, for the D6 SEM ranges Developer (3.87–4.71) and MSc (2.90–3.70) do not overlap, as the Developer lower bound exceeds the MSc upper bound. This non-overlap suggests a potentially statistically meaningful difference between the two cohorts.

Emotional Markers (**D7**) received an overall mean of 3.82 (Agree), with developers (M = 3.71) and MSc participants (M = 3.90) agreeing that the chosen emotional markers accurately represent the emotional states associated with IDE interaction. This supports the emotional mapping used in the dataset as representative of authentic developer experiences. Finally, for D7, the SEM intervals Developer (3.24–4.18) and MSc (3.67–4.13) overlap, suggesting no statistically distinct difference in precision-adjusted estimates.

On the whole, participants reported positive perceptions across all seven dimensions (D1–D7), indicating that the TIDE dataset was generally perceived as clear, effective, and practically valuable. Descriptive comparisons of the mean scores and associated SEM intervals suggest broadly similar response patterns between developers and MSc participants, with developers appearing to express somewhat stronger confidence in the dataset's integration potential and usefulness for reducing technostress. These observations should be interpreted descriptively rather than inferentially.

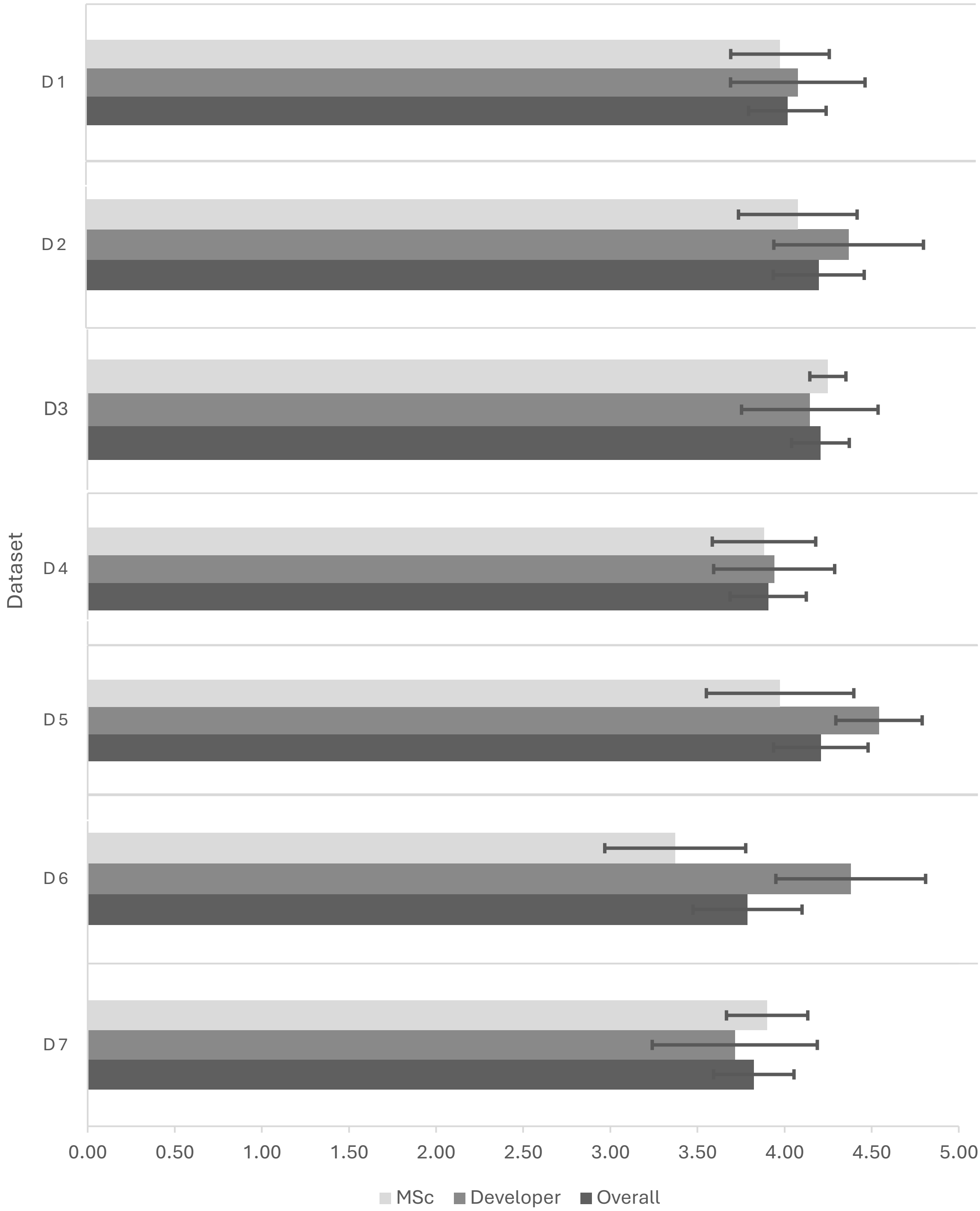


Figure 5.5: Dataset Evaluation Scores

5.6.1.1 Internal Consistency

The seven-item scale assessing the TIDE biometric dataset (Clarity, Effectiveness, Practicality, Integration, Technostress Reduction, Confidence, and Emotional Markers) produced a Cronbach's Alpha of $\alpha = 0.77$, indicating strong internal reliability. This higher coefficient suggests that participants were associated with greater consistency in their evaluation of the dataset's clarity, structure, and research utility.

Unlike the framework scale, the dataset evaluation items were more conceptually cohesive, focusing primarily on a single construct, perceived dataset quality and applicability. This coherence likely contributed to stronger inter-item correlations, as participants evaluated concrete characteristics such as the dataset's structure, the inclusion of emotional and behavioural markers, and its potential for research reuse. In other words, participants were judging a tangible artefact rather than a conceptual process, which tends to produce higher consistency across responses [379].

Furthermore, the dataset's measurable nature and the presence of clearly defined attributes (e.g., phase segmentation, emotional metrics, clarity of documentation) allowed for more objective evaluation, reducing interpretive ambiguity between participants. Both developers and MSc students appeared to share a common understanding of what constituted dataset clarity or utility, resulting in stronger reliability.

The difference between the framework ($\alpha = 0.57$) and dataset ($\alpha = 0.77$) alpha values also highlights an important methodological insight: participants tend to provide more stable and convergent feedback when evaluating concrete artefacts rather than abstract conceptual models. In this sense, the dataset reliability supports the study's broader approach, integrating both conceptual and empirical contributions to achieve a holistic understanding of technostress-aware design.

5.6.2 Qualitative Analysis

The same descriptive lexical analysis approach outlined in Section 5.5.2 was applied to the dataset evaluation responses. A word cloud was generated to provide an overview of the terminology most frequently used by participants when discussing the TIDE dataset. As previously noted, the visualisation is intended as descriptive lexical evidence that complements the quantitative findings and should not be interpreted as a formal qualitative-analysis method or as a direct measure of the importance of individual concepts.

5.6.2.1 Lexical Divergence

The lexical divergence between cohorts reflects a fundamental difference in how the biometric dataset itself is conceptualised. Developers (Figure 5.6) predominantly treat the biometric dataset as a technical artefact to be engineered, captured, and optimised. The dataset is framed as a structured data pipeline, something to be collected, processed, and leveraged for insight generation. The primary concern appears to be measurement fidelity and functional implementation rather than interpretive or ethical implications.

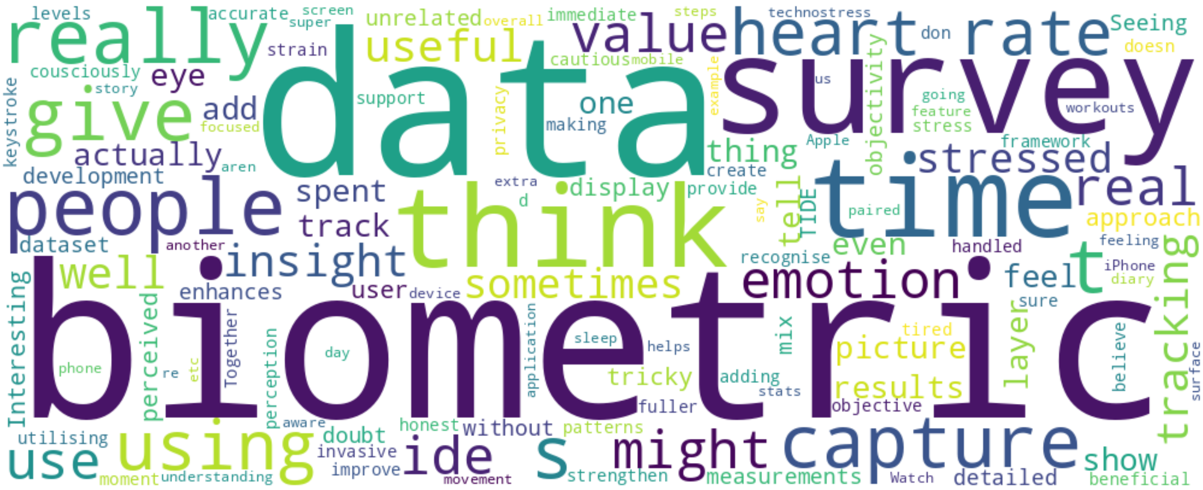


Figure 5.6: Developer Dataset Word Cloud

In contrast, MSc participants treat the biometric dataset as a socio-technical construct that must be valid, secure, and emotionally representative. Their lexical emphasis (Figure. 5.7) on emotion, stress, security, realism, and correctness suggests that the dataset is evaluated in terms of its interpretive legitimacy and psychological accuracy. Rather than focusing solely on how data are captured, this cohort foregrounds whether the biometric signals meaningfully reflect real emotional states and whether such data collection is ethically and securely managed. The dataset is therefore framed not merely as a technical resource, but as a sensitive representation of human affect that requires validation and trust.

Figure 5.7: MSc Dataset Word Cloud

The distinction is thus conceptual rather than topical. Both cohorts discuss the same biometric dataset, yet developers approach it as an instrumented technical system, whereas MSc participants approach it as an emotionally and ethically consequential representation of human experience. This difference reflects expertise-dependent cognitive framing and suggests that domain familiarity influences whether biometric data are perceived primarily as engineering artefacts or as socio-emotional evidence.

5.6.2.2 Sample Responses

5.6.2.2.1 Developer

The developer feedback (for **D8**) showed strong support for the inclusion of biometric data as an additional source of insight alongside qualitative and quantitative survey data. Many participants recognised its potential to provide a more complete and objective view of user experience and technostress during IDE use.

Several developers viewed the approach positively.

> *P1 describing it as an "interesting approach, I like it".*
>
> *P3, highlighted that "biometric data without a doubt adds objectivity to the mix," explaining that because the "TIDE dataset has detailed biometric measurements (real-time emotion), this enhances the survey data as it captures what people might not consciously recognise."*
>
> *Similarly, P13 supported this idea, stating that "using biometric data would add real value, as it gives an objective layer to support the survey results," noting that while "surveys capture perception, biometrics like keystroke patterns or heart rate provide a more immediate picture of stress levels."*

Developers also reflected on how biometrics could uncover what participants might not be aware of in the moment.

> *P17 explained that "using biometric data like tracking heart rate or eye movement can actually be super useful when paired with survey results," because "surveys tell us what people think or say they're feeling, but biometrics can show what's going on under the surface." They further compared it to "having a diary and an Apple Watch to track our steps or workouts or sleep — one tells a story, the other one shows the stats."*

Some participants focused on the conditions necessary for this approach to be effective.

> *P8 mentioned that biometric data "can improve insight if tracking and utilising the data well, so I believe it is beneficial."*
>
> *P10 agreed with the potential benefits, saying, "I think adding biometric data could be really useful, because sometimes surveys don't capture how stressed or tired you actually feel in the moment." However, they also raised a valid concern: "The only thing I'd be cautious about is privacy and making sure it doesn't feel invasive."*

A few participants did express caution about interpreting biometric data in isolation.

> *P2 noted that "the perceived value might be tricky as a user can display emotions during development unrelated to IDE use."*

This suggests an awareness that emotional or physiological responses may not always be directly tied to the development task at hand, reinforcing the importance of context when analysing biometric signals.

Developers agreed that integrating biometric data provides an additional, more objective dimension to understanding technostress and user experience. While surveys reveal how users perceive their stress or engagement, biometric data shows how they actually respond in real time. When combined, these sources create a richer and more balanced perspective, strengthening the reliability and depth of insights produced by the framework.

Developer feedback in the final question (for **D9**) showed strong encouragement for the research and its potential to improve developer experience in the future.

> *P10 said, "it's a really interesting piece of research, and I can see how it could make a real difference for developers. Looking forward to seeing how the results shape future IDEs."*

This reflected a positive attitude toward the study's relevance and how its findings could influence the next generation of developer tools.

> *P17 provided a thoughtful comment about the role of biometric data in the research. They said, "I'd just add that biometric data shouldn't take over or replace what people actually say about their experiences, it should work alongside it. At the end of the day, people's voices matter most. The point isn't to track or monitor them, but to better understand what they're going through and find ways to improve things. If it's used with care and respect, this kind of data could help build tools that really get people not just what they do, but how they feel."*

This highlights the importance of balance, showing that while biometric data can give valuable insight, it should complement rather than replace personal feedback.

Developers viewed the research as meaningful and forward-thinking. They valued its potential to improve future IDE design and workplace wellbeing, while also stressing the importance of maintaining a human focus where both data and personal experience work together to inform better, more empathetic tools.

5.6.2.2.2 Postgraduate (MSc)

MSc students generally saw the use of biometric data as a valuable addition to the framework (for **D8**), recognising its ability to provide more accurate and detailed insights into technostress and user experience. Many participants highlighted that biometric data could reveal information that surveys might overlook, particularly in capturing real-time emotional and physiological responses.

> *P4 explained that "using biometric data can give a more accurate picture of technostress because it shows real-time physical responses that surveys might miss." They added that it "can complement qualitative and quantitative feedback, helping identify stress patterns more precisely," while also warning that "privacy and consent should be carefully managed."*
>
> *Similarly, P5 shared that "combining the data and the result can give a deeper insight as it captures reaction people may not consciously report," but also noted that it would "still need a strong ethical safeguard."*

Several participants pointed out that biometric data helps to overcome the limits of traditional self-reporting.

> *P7 said that "using biometric data is a powerful approach because quantitative/qualitative data can be limited by recall bias."*
>
> *P14 also expressed support, explaining that "biometric data adds deeper insights that surveys alone might miss and it helps capture real emotional responses and makes the results more reliable."*
>
> *Likewise, P16 noted that "it's very useful as capturing emotions will give insights into relevant data and experiences."*

A few participants provided more specific reflections on how biometrics could enrich the framework.

> *P11 described that "using biometric data adds a really interesting phase to the survey insights," explaining that it "shows real time emotions and attention patterns, which can help link user stress to actual performance."*

However, some participants also raised considerations and limitations.

> *P9 commented that "I kinda love the concepts of emotion but still there are more biometrics data which is further the differences even more, and the data is quite subjective so it depends."*
>
> *P15 pointed out that "honestly it's not detailed enough," adding that "there are definitely elements missing in terms of the specifics of expression that would rely on cultural and background understandings."*
>
> *P12 also cautioned that "it's a good method but need to care about the privacy policy."*

As a whole, MSc students viewed biometric data as an important step toward gaining a more complete picture of technostress and emotional experience in development environments. They agreed that combining biometric measures with survey data would give deeper, more reliable insight into user stress and behaviour. At the same time, they emphasised the need for careful handling of privacy, ethical safeguards, and cultural differences to ensure that biometric analysis remains both respectful and meaningful.

MSc students shared positive final thoughts (for **D9**) about the research and its potential impact.

> *P4 said, "overall, combining surveys with biometric data in a technostress-aware framework could make stress detection more accurate and personalised." They added that "the key is to keep the approach user-friendly, respect privacy, and ensure the insights lead to practical, non-intrusive improvements."*

This showed support for the framework's direction while reminding that its success depends on maintaining usability and ethical care.

> *P7 commented that "it's an excellent project, very refreshing," reflecting a positive overall impression of the study and its aims.*

MSc participants saw the research as an important step forward. They recognised that combining survey and biometric data can make stress detection more meaningful and tailored, as long as the system remains accessible, respectful, and focused on creating practical improvements for users.

5.6.3 Addressing Research Question Two

Question: How do developers and MSc User Experience Design students evaluate the research utility, credibility, and future potential of the TIDE dataset as a complementary resource to the framework? (RQ2)

Findings: Both the quantitative and qualitative evaluations suggest a consistently positive perception of the TIDE biometric dataset across developers and MSc User Experience Design students. Overall mean scores for all evaluated dimensions were above the neutral midpoint, indicating strong agreement about the dataset's clarity, structure, and research value. Participants recognised it as a relevant and practically useful resource for advancing technostress-aware and technostress-oriented IDE research.

Developers generally rated the dataset slightly higher than MSc participants, reflecting their stronger familiarity with data-driven workflows and their ability to visualise how such a resource could be applied in real-world development contexts. They valued its potential to support fine-tuning of IDE features and saw the biometric components as powerful tools for identifying emotional and cognitive states during programming. MSc participants also viewed the dataset positively, highlighting its role in providing transparency and objectivity to complement traditional survey data, while emphasising the importance of maintaining ethical standards and privacy when using biometric measures.

The qualitative findings reinforced these results, showing strong support from both groups for integrating biometric data as an additional layer of insight alongside qualitative and quantitative feedback. Participants agreed that combining these data types allows for a richer and more accurate understanding of technostress, bridging the gap between self-reported perception and physiological response. The TIDE dataset was seen as enhancing the credibility of the framework by enabling triangulation between perceived, behavioural, and physiological indicators, a key step toward validating the emotional and cognitive dimensions of programming. Participants acknowledged that the dataset's integration into the framework creates new opportunities for adaptive, data-driven IDE design. Developers, in particular, expressed strong confidence in its ability to inform technostress-aware enhancements, recognising how real-time emotional markers could be used to dynamically adjust the development environment to reduce stress and improve focus. MSc students supported this view, seeing the dataset as a foundation for more human-centred, emotionally intelligent systems capable of evolving through continuous learning.

Both groups also discussed the dataset's broader research utility and replicability. The clear documentation of variables and emotion mappings was seen as a strength, ensuring that future researchers could reuse and extend the dataset for comparative or longitudinal studies. By balancing objectivity with ethical responsibility, participants felt that the TIDE dataset holds substantial potential to advance both academic inquiry and applied in technostress mitigation and affective computing.

Discussion: Developers and MSc participants perceived the TIDE dataset as a credible, clear, and practically valuable resource that complements the framework effectively. Developers viewed it as highly applicable for real-world integration and feature refinement, while MSc participants emphasised its role in enhancing research accuracy and ethical transparency. Together, their feedback supports that the TIDE dataset provides a robust foundation for understanding and mitigating technostress through a balanced combination of objective biometric data and subjective user insight. Its integration with the framework represents a significant advancement toward creating adaptive, human-centred, and emotionally responsive IDEs capable of supporting both wellbeing and outcomes.

5.7 Summary

Collectively, findings across both eh framework and dataset provide preliminary but converging evidence that the framework and dataset function effectively as complementary components of a broader technostress-oriented research and design approach, combining structured intervention design with multi-modal empirical insight.

Building on the evaluation presented in this chapter, which assessed the proposed framework and dataset through developer and MSc student perspectives the subsequent Chapter 6 provides the final conclusion of the thesis. It synthesises the findings across all chapters, reflecting on how the taxonomy, framework, and TIDE dataset collectively contribute to understanding and addressing technostress in IDE environments. Outlining the overall contributions of the research, discusses its limitations, and identifies directions for future work in technostress-aware system design and evaluation.

Chapter *Six*

6 Conclusion

Findings suggest that technostress in computing education should not be viewed solely as a psychological phenomenon, but as a design challenge that can be influenced through the tools students use on a daily basis. The taxonomy established a structured understanding of the relationships between technostress, assistive technology, and educational contexts, identifying key gaps in the literature and highlighting the limited attention given to practical interventions within programming environments. Building upon these findings, the framework translated theoretical concepts into an actionable process for identifying, implementing, and evaluating IDE modifications informed by user feedback and technostress principles. The TIDE dataset subsequently provided an empirical resource through which emotional, behavioural, and usability-related responses could be examined within the context of IDE interaction.

Taken together, these contributions form a coherent progression from understanding the problem, to designing interventions, to generating evidence that can support future research and development. Rather than representing three independent outputs, the taxonomy, framework, and dataset are intended to function as complementary components that support the development of more human-centred and technostress-aware programming environments.

The findings provide preliminary evidence that IDE modifications informed by user feedback may contribute to improvements in usability, user experience, and perceptions of technostress. However, the thesis does not claim to prove causal relationships between specific IDE features and reductions in technostress. The exploratory nature of the study, the modest sample sizes, and the contextual limitations of the experimental environment mean that the findings should be interpreted as indicative rather than definitive. Additional studies involving larger and more diverse populations, longer-term deployments, and alternative development environments would be required to establish the generalisability of the proposed approach.

Despite these limitations, the work has several broader implications. For IDE designers, the findings highlight the importance of incorporating adaptive, personalisable and user-centred features that respond to both functional and technostress-related needs. For computing educators, the research suggests that programming tools may influence not only learning outcomes but also emotional experiences and perceptions of stress during development activities. For technostress researchers, the thesis demonstrates the value of combining qualitative feedback, usability measures, and biometric indicators to obtain a more holistic understanding of user experience in technology-intensive environments.

Ultimately, this thesis contributes towards a growing shift from viewing software development tools solely as productivity instruments towards recognising them as environments that can shape cognitive, emotional, and behavioural experiences. In doing so, it provides foundations for future research into emotionally conscious, wellbeing-oriented, and technostress-aware software development environments.

6.1 Summary of Contributions

6.1.1 Addressing Research Question One

Question: How can technostress and its influencing factors be systematically studied and categorised?

Findings: Refer to **Chapter Two**

Discussion: This chapter conducts a systematic literature review to systematically study technostress and its influencing factors. This review has resulted in a systematically derived taxonomy which categorises technostress factors and how they can be measured. The derived metrics of performance, productivity, and satisfaction, identified through the taxonomy, offer measurable indicators for assessing the effectiveness of technostress-related interventions. The systematic literature review also uncovers the main gap being addressed in the remainder of the thesis: in educational contexts (specifically IDEs) technostress-awareness is not captured systematically in the design process. This gap is thereafter addressed in this research through contributions that systematically identify and mitigate technostress triggers while supporting emotional wellbeing.

6.1.2 Addressing Research Question Two

Question: How can classical software design and evaluation practices be adapted into a feedback-driven, technostress-aware framework that guides the development of IDEs supporting technostress reduction and mental wellbeing?

Findings: Refer to **Chapter Three**

Discussion: Classical software design practices can be effectively adapted into a technostress-aware, feedback-driven framework by embedding user feedback loops and emotional evaluation mechanisms (grounded in the systematically derived metrics established in the technostress taxonomy introduced in Chapter 2) throughout the design lifecycle. The TIDEs framework evidences how the framework supports iterative IDE refinement by following the structured three stages of development: Preparation and Design, where scope needs are defined; Participant Selection and Validation, which ensures appropriate and reliable user input; and Iterative Evaluation and Refinement, where behavioural and psychological evidence is used to guide evidence-based design decisions rather than assumption-driven changes.

6.1.3 Addressing Research Question Three

Question: How can a reusable emotional dataset, captured during IDE-based programming tasks, be constructed and analysed to inform technostress-centric user interface design and decision making?

Findings: Refer to **Chapter Four**

Discussion: In practical terms, the creation and expert validation of the TIDE dataset evidence the credibility and research utility of biometric data as a complementary insight mechanism to traditional survey-based methods. It establishes a reusable foundation for future studies in affective computing and software engineering, allowing for cross comparisons, longitudinal analysis, and adaptive IDE development informed by real-time user emotion. As part of this contribution, a representative subset of emotional markers was refined to ensure an appropriate balance between analytical scope and practical implementation. The resulting set provides coverage of key affective states relevant to IDE-based programming activity while maintaining manageable analytical complexity and supporting consistent emotional tracking

6.1.4 Addressing Research Question Four

Question: How can the proposed framework and dataset be evaluated in an end-to-end manner through expert assessment?

Discussion: Refer to **Chapter Five**

Result: The evaluation phase involving both professional developers and MSc User Experience Design participants, indicate that the proposed technostress-aware framework and the accompanying TIDE biometric dataset were perceived as credible, relevant, and complementary resources. The combined evaluation evidence that the framework and TIDE dataset can effectively complement each other to provide an evidence-based approach for understanding and mitigating technostress in programming environments.

6.2 Threats to Validity

6.2.1 Internal

Internal Threat (IT) validity concerns the extent to which the observed outcomes in this research can be confidently attributed to the experimental conditions and framework design rather than uncontrolled factors.

Alignment Across Data Sources: While triangulating biometric, qualitative, and quantitative data increased the reliability of the interpretations, some risk of bias remains in aligning these sources, particularly where expected outcomes could influence analysis. To mitigate this risk, statistical analyses were conducted at each evaluation stage to support objective interpretation and ensure that conclusions were grounded in empirical evidence rather than subjective alignment across data sources. This approach helped balance qualitative insights with quantitative validation, strengthening the robustness of the evaluation outcomes.

Qualitative Analysis Approach: The qualitative analysis adopted an exploratory and descriptive approach based on researcher-led thematic interpretation supported by lexical visualisation techniques. While sample quotations were included to improve transparency and evidentiary grounding, formal thematic-analysis procedures involving independent coders or inter-rater reliability assessment were not conducted. Consequently, identified themes should be interpreted as indicative rather than definitive representations of participant perspectives.

6.2.2 External

External Threat (ET) validity refers to the extent to which the findings of this study can be generalised beyond the controlled research setting.

IDE Specificity: The proposed framework and supporting dataset were evaluated exclusively within the context of VSCode. While VSCode represents a widely adopted industry-standard IDE among both students and professional developers, its interface structure, extension ecosystem, and interaction patterns may differ substantially from those of other environments. As a result, the applicability of the findings to alternative IDEs may be limited. Differences in workflow design, feedback mechanisms, and customisation models could influence how technostress manifests and how effectively the framework transfers across platforms. Future work should therefore validate the framework across a broader range of development tools and IDEs to further strengthen the generalisability and reliability of the constructs examined.

6.2.3 Construct

Construct Validity (CV) concerns whether the study accurately measures what it intends to measure, ensuring that the instruments, metrics, and datasets genuinely capture the constructs of technostress, wellbeing, and user experience.

Participant: Construct validity was further strengthened through a multi-layered participant design incorporating both end-user and expert perspectives. A total of 37 participants contributed to the study: 20 undergraduate computing students who completed two experimental cycles using the baseline and modified IDE, and 17 MSc students and professional developers who evaluated the framework and the biometric dataset. This design ensured that constructs were examined from both practical usage and expert evaluation perspectives. Nevertheless, whilst the modest sample size design offers strong internal consistency and relevance to the research objectives, it also limits the robustness of subgroup comparisons and may constrain the statistical sensitivity when examining variations in construct interpretation across participant roles. Future studies could expand the sample size and incorporate longitudinal validation. Studies that collect direct biometric or affective measures in HCI and software interaction frequently operate with small to moderate sample sizes due to practical constraints in acquiring high-quality physiological data and the exploratory nature of the research. In the present study, this constraint is addressed through a two-cycle within-subject experimental design, in which the same cohort of 20 computing students participated across both the baseline and modified IDE conditions. Rather than increasing participant numbers at the expense of data depth and control, this design enables direct comparison of IDE-induced technostress and user experience within individuals, reducing inter-participant variability and strengthening internal validity.

Prior work supports that modest samples are both typical and methodologically appropriate in comparable contexts. For example, [331] conducted a biometric investigation with 16 professional software testers, collecting EEG, electrodermal activity, and ET data during software testing tasks. Similarly, [380] examined emotional responses to software interfaces using facial EMG and performance measures with 10 participants, reflecting standard practice in early-stage affective computing research. Studies examining programmer behaviour further reinforce this norm. The work in [329] employed EEG and eye-tracking measures with 37 participants to study programming behaviour, illustrating that even at larger scales, biometric IDE studies remain constrained by experimental and data-quality demands. In related usability-focused research, [381] evaluated an inclusive multimodal voice-based code navigation system with 14 developers in the main study and five participants in a preliminary phase, producing actionable insights despite the modest sample size. Research centred on student populations and IDE interaction also commonly adopts smaller cohorts. For instance, [382] investigated IDE experiences of students with ADHD using think-aloud methods with nine participants, while [383] conducted a usability evaluation of a block-based coding environment with 14 participants, demonstrating that focused programming usability studies can validly assess efficacy, efficiency, and satisfaction with participant sizes below 20.

Within this methodological landscape, the two-cycle design compensates for the modest sample size by enabling repeated-measures analysis across contrasting IDE configurations, increasing sensitivity to design-induced changes in technostress, emotional response, and task performance. This approach aligns with established practice in biometric and affective HCI research, where depth of measurement, experimental control, and within-subject comparison are prioritised over large-scale sampling.

6.2.4 Ecological

Ecological Validity (EV) refers to how well the experimental conditions reflect real-world environments, determining the extent to which the observed behaviours and emotional responses can be generalised to authentic user contexts.

Contextual Realism: Although the programming tasks were designed to be representative of realistic IDE usage and aligned with students' expected skill levels, they were conducted within a time-bounded, controlled experimental context. This setting may not fully capture the complexity, interruptions, prolonged cognitive load, or accountability pressures associated with real coursework deadlines or professional development projects. As a result, the intensity and persistence of technostress experienced during the study may differ from that encountered in naturalistic settings. Future work could strengthen ecological validity by embedding the framework into longer-term academic modules or live development workflows, allowing emotional and behavioural responses to emerge under more authentic conditions.

6.3 Limitations and Future Direction

Beyond the thesis, there is a clear intention to take this work further and continue developing the concepts and tools introduced in this study. While the proposed framework indicates strong internal consistency and relevance to the research objectives, several limitations are acknowledged that inform both the interpretation of the contributions and directions for future research.

6.3.1 Increase Demographic Range

Future research should involve a larger and more diverse participant cohort spanning multiple universities, educational levels, and disciplinary backgrounds to improve the generalisability of the framework and its outcomes. Broadening the demographic scope would enable the capture of a wider range of experiences with IDE use and technostress, strengthening the applicability of the findings across educational contexts. In addition, future studies should incorporate pre-assessment of baseline stress and wellbeing levels prior to experimental participation. Accounting for individual differences in stress susceptibility may provide deeper insight into how technostress manifests during IDE interaction and whether existing stress levels moderate emotional, behavioural, or performance-related outcomes. Finally, this study did not explicitly examine the experiences of neurodivergent or disabled students, whose interaction with development environments may differ substantially from neurotypical users. Including these groups in future work would support more inclusive design considerations and help ensure that technostress-aware IDE frameworks address accessibility, cognitive diversity, and varied support needs more effectively.

6.3.2 Involve Developers in the Experimental Phase

While professional developers and MSc participants contributed valuable expert feedback on the framework and dataset, future work could strengthen this contribution by involving developers more explicitly upstream in the framework and dataset development process, rather than as an additional end-user group in experimental evaluations. In particular, experienced developers could be engaged during the refinement of framework stages, task design, and metric selection to ensure alignment with diverse software development lifecycles and real-world IDE practices. Such involvement would help validate that the dataset functions as a meaningful benchmark for evaluating technostress-aware IDE adaptations and that the framework remains applicable beyond academic settings. This expert-led integration would complement, rather than replace, student-based experimental studies, supporting both ecological validity and transferability to professional contexts

6.3.3 Technical Development

A practical next step is to develop the proposed framework as a VSCode extension available on the marketplace. This extension could capture real-time behavioural, interaction, and biometric data during programming activities and use this information to adapt the IDE environment dynamically at runtime. By continuously monitoring user interactions and affective indicators, the extension could provide adaptive recommendations, interface adjustments, and context-aware support mechanisms aimed at reducing technostress and improving user experience during everyday coding activities. Over time, iterative user feedback and continued data collection could further refine the system, improving its accuracy, responsiveness, and personalisation capabilities.

Further technical development of the extension could explore how user expression data, particularly facial and affective indicators captured during interaction, can support adaptive and emotionally aware IDE functionality. These approaches would use emotional signals as evaluative and diagnostic inputs to dynamically modify interface behaviour and user support in response to detected cognitive or emotional states.

Emotion-Informed Interface Evaluation: The extension could continuously monitor interaction patterns and emotional responses alongside traditional usability metrics. Recurrent expressions associated with frustration, confusion, or disengagement during interaction with specific IDE components could help identify problematic interface areas and automatically trigger adaptive recommendations or interface refinements.

Reduction of Cognitive Load Through Affective Signals: The extension could detect moments where negative affect corresponds with high interaction density or complex configuration options. In response, the IDE could dynamically simplify navigation structures, reduce visual clutter, prioritise essential tools, or temporarily suppress non-critical notifications during cognitively demanding interactions.

Phase-Aware Interface Support: Emotional responses often vary across task phases. The extension could identify whether the user is in the analysis, implementation, or testing stage and adapt system behaviour accordingly. This may include providing contextual hints, enhanced debugging support, instructional cues, or clearer feedback mechanisms during high-stress development phases while minimising interruptions during periods of sustained engagement or productivity.

Benchmarking Design Alternatives: The extension could collect anonymised emotional and interaction data across different interface configurations to evaluate how alternative IDE designs affect user emotional stability, engagement, and stress levels. This would support evidence-based interface optimisation using real-world behavioural feedback.

Inclusive and Wellbeing-Oriented Design: Sustained negative expressions or stress-related interaction patterns detected by the extension may indicate that certain interface components impose excessive emotional or cognitive demands on users. The system could therefore introduce adaptive accessibility features, personalised layouts, adjustable interface complexity, or customisable workflows tailored to individual user needs and emotional responses.

7 References

[1] Jenny Bermingham, 'Non-continuation: UK Performance Indicators', HESA, 2022. [Online]. Available: https://www.hesa.ac.uk/data-and-analysis/performance-indicators/non-continuation

[2] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*, 1st edn. CRC Press, 2018. doi: 10.1201/9780203736166.

[3] Shneiderman, 'Direct Manipulation: A Step Beyond Programming Languages', *Computer*, vol. 16, no. 8, pp. 57–69, Aug. 1983, doi: 10.1109/MC.1983.1654471.

[4] D. A. Norman, *The design of everyday things*, Revised and Expanded ed. New York: Basic Books, 2013.

[5] R. W. Picard, *Affective Computing*. The MIT Press, 1997. doi: 10.7551/mitpress/1140.001.0001.

[6] R. Cowie *et al.*, 'Emotion recognition in human-computer interaction', *IEEE Signal Process. Mag.*, vol. 18, no. 1, pp. 32–80, Jan. 2001, doi: 10.1109/79.911197.

[7] R. A. Calvo and S. D'Mello, 'Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications', *IEEE Trans. Affect. Comput.*, vol. 1, no. 1, pp. 18–37, Jan. 2010, doi: 10.1109/T-AFFC.2010.1.

[8] C. Brod, *Technostress: the human cost of the computer revolution*. Reading, Mass: Addison-Wesley, 1984.

[9] G. La Torre, V. De Leonardis, and M. Chiappetta, 'Technostress: how does it affect the productivity and life of an individual? Results of an observational study', *Public Health*, vol. 189, pp. 60–65, Dec. 2020, doi: 10.1016/j.puhe.2020.09.013.

[10] T. Fischer, A. Pehböck, and R. Riedl, *Is the Technostress Creators Inventory Still an Up-To-Date Measurement Instrument? Results of a Large-Scale Interview Study*. 2019.

[11] A. Chrysikos and N. Bamford, 'Retention in Second Year Computing Students in a London-Based University During the Post-COVID-19 Era Using Learned Optimism as a Lens: A Statistical Analysis in R', in *Proceedings of Data Analytics and Management*, vol. 785, A. Swaroop, Z. Polkowski, S. D. Correia, and B. Virdee, Eds, in Lecture Notes in Networks and Systems, vol. 785. , Singapore: Springer Nature Singapore, 2024, pp. 57–69. doi: 10.1007/978-981-99-6544-1_5.

[12] M. Asghar and A. Minichiello, 'Two sides to every psyche: Implications of positive psychology for "mental health" research in engineering education', *J. Eng. Educ.*, vol. 112, no. 1, pp. 12–17, Jan. 2023, doi: 10.1002/jee.20493.

[13] A. V. Horwitz and T. L. Scheid, *A handbook for the study of mental health: Social contexts, theories, and systems.* in A handbook for the study of mental health: Social contexts, theories, and systems. New York, NY, US: Cambridge University Press, 1999, pp. xvii, 676.

[14] H. Herrman, S. Saxena, and R. Moodie, *Promoting mental health: concepts, emerging evidence, practice; a report*. Geneva: World Health Organization, 2005.

[15] S. M. A. Lamers, G. J. Westerhof, E. T. Bohlmeijer, P. M. Ten Klooster, and C. L. M. Keyes, 'Evaluating the psychometric properties of the mental health Continuum-Short Form (MHC-SF)', *J. Clin. Psychol.*, vol. 67, no. 1, pp. 99–110, Jan. 2011, doi: 10.1002/jclp.20741.

[16] E. L. Deci and R. M. Ryan, 'Hedonia, eudaimonia, and well-being: an introduction', *J. Happiness Stud.*, vol. 9, no. 1, pp. 1–11, Jan. 2008, doi: 10.1007/s10902-006-9018-1.

[17] E. Diener, 'Subjective well-being.', *Psychol. Bull.*, vol. 95, no. 3, pp. 542–575, 1984, doi: 10.1037/0033-2909.95.3.542.

[18] A. S. Waterman, 'Two conceptions of happiness: Contrasts of personal expressiveness (eudaimonia) and hedonic enjoyment.', *J. Pers. Soc. Psychol.*, vol. 64, no. 4, pp. 678–691, Apr. 1993, doi: 10.1037/0022-3514.64.4.678.

[19] A. S. Waterman, S. J. Schwartz, and R. Conti, 'The Implications of Two Conceptions of Happiness (Hedonic Enjoyment and Eudaimonia) for the Understanding of Intrinsic Motivation', *J. Happiness Stud.*, vol. 9, no. 1, pp. 41–79, Jan. 2008, doi: 10.1007/s10902-006-9020-7.

[20] S. Galderisi, A. Heinz, M. Kastrup, J. Beezhold, and N. Sartorius, 'Toward a new definition of mental health', *World Psychiatry*, vol. 14, no. 2, pp. 231–233, Jun. 2015, doi: 10.1002/wps.20231.

[21] G. E. Vaillant, 'Positive mental health: is there a cross-cultural definition?', *World Psychiatry*, vol. 11, no. 2, pp. 93–99, Jun. 2012, doi: 10.1016/j.wpsyc.2012.05.006.

[22] S. K. Lipson *et al.*, 'Trends in college student mental health and help-seeking by race/ethnicity: Findings from the national healthy minds study, 2013–2021', *J. Affect. Disord.*, vol. 306, pp. 138–147, Jun. 2022, doi: 10.1016/j.jad.2022.03.038.

[23] D. Hammoudi Halat, A. Soltani, R. Dalli, L. Alsarraj, and A. Malki, 'Understanding and Fostering Mental Health and Well-Being among University Faculty: A Narrative Review', *J. Clin. Med.*, vol. 12, no. 13, p. 4425, Jun. 2023, doi: 10.3390/jcm12134425.

[24] D. G. Lannin, D. L. Vogel, and P. J. Heath, ‘Can reflecting on personal values online increase positive beliefs about counseling?’, *J. Couns. Psychol.*, vol. 64, no. 3, pp. 261–268, Apr. 2017, doi: 10.1037/cou0000201.

[25] S. Folkman, ‘Stress: Appraisal and Coping’, in *Encyclopedia of Behavioral Medicine*, M. D. Gellman and J. R. Turner, Eds, New York, NY: Springer New York, 2013, pp. 1913–1915. doi: 10.1007/978-1-4419-1005-9_215.

[26] S. Folkman and R. S. Lazarus, ‘Coping as a mediator of emotion.’, *J. Pers. Soc. Psychol.*, vol. 54, no. 3, pp. 466–475, Mar. 1988, doi: 10.1037/0022-3514.54.3.466.

[27] A. E. Jeffers, ‘The Field of Computing Needs to Take Care of Its Mental Health’, *Comput. Sci. Eng.*, vol. 24, no. 2, pp. 91–94, Mar. 2022, doi: 10.1109/MCSE.2022.3162034.

[28] Marta Chiappetta, ‘The Technostress: definition, symptoms and risk prevention’, *Senses Sci.*, 2017, doi: 10.14616/sands-2017-1-358361.

[29] T. S. Ragu-Nathan, M. Tarafdar, B. S. Ragu-Nathan, and Q. Tu, ‘The Consequences of Technostress for End Users in Organizations: Conceptual Development and Empirical Validation’, *Inf. Syst. Res.*, vol. 19, no. 4, pp. 417–433, Dec. 2008, doi: 10.1287/isre.1070.0165.

[30] J. Galvin *et al.*, ‘Technostress, coping, and anxious and depressive symptomatology in university students during the Covid-19 pandemic’, *Eur. J. Psychol.*, vol. 18, no. 3, pp. 302–318, Aug. 2022, doi: 10.5964/ejop.4725.

[31] Ayyagari, Grover, and Purvis, ‘Technostress: Technological Antecedents and Implications’, *MIS Q.*, vol. 35, no. 4, p. 831, 2011, doi: 10.2307/41409963.

[32] A. Chuang, C.-T. Shen, and T. A. Judge, ‘Development of a Multidimensional Instrument of Person-Environment Fit: The Perceived Person-Environment Fit Scale (PPEFS): Multidimensional Instrument of Person-Environment Fit’, *Appl. Psychol.*, vol. 65, no. 1, pp. 66–98, Jan. 2016, doi: 10.1111/apps.12036.

[33] M. Tarafdar, Q. Tu, B. S. Ragu-Nathan, and T. S. Ragu-Nathan, ‘The Impact of Technostress on Role Stress and Productivity’, *J. Manag. Inf. Syst.*, vol. 24, no. 1, pp. 301–328, Jul. 2007, doi: 10.2753/MIS0742-1222240109.

[34] P. Upadhyaya and Vrinda, ‘Impact of technostress on academic productivity of university students’, *Educ. Inf. Technol.*, vol. 26, no. 2, pp. 1647–1664, Mar. 2021, doi: 10.1007/s10639-020-10319-9.

[35] N. Dragano and T. Lunau, ‘Technostress at work and mental health: concepts and research results’, *Curr. Opin. Psychiatry*, vol. 33, no. 4, pp. 407–413, Jul. 2020, doi: 10.1097/YCO.0000000000000613.

[36] G. Nimrod, ‘Technostress in a hostile world: older internet users before and during the COVID-19 pandemic’, *Aging Ment. Health*, vol. 26, no. 3, pp. 526–533, Mar. 2022, doi: 10.1080/13607863.2020.1861213.

[37] E. Ingusci *et al.*, ‘Workload, Techno Overload, and Behavioral Stress During COVID-19 Emergency: The Role of Job Crafting in Remote Workers’, *Front. Psychol.*, vol. 12, p. 655148, Apr. 2021, doi: 10.3389/fpsyg.2021.655148.

[38] I. O. Panisoara, I. Lazar, G. Panisoara, R. Chirca, and A. S. Ursu, ‘Motivation and Continuance Intention towards Online Instruction among Teachers during the COVID-19 Pandemic: The Mediating Effect of Burnout and Technostress’, *Int. J. Environ. Res. Public. Health*, vol. 17, no. 21, p. 8002, Oct. 2020, doi: 10.3390/ijerph17218002.

[39] S. A. Murray, H. D. Shuler, J. S. Davis, E. C. Spencer, and A. O. Hinton Jr, ‘Managing technostress in the STEM world’, *Trends Biotechnol.*, vol. 40, no. 8, pp. 903–906, Aug. 2022, doi: 10.1016/j.tibtech.2022.05.001.

[40] D. Sommer, E. Lermer, F. Wahl, and L. I. Lopera G., ‘Assistive technologies in healthcare: utilization and healthcare workers perceptions in Germany’, *BMC Health Serv. Res.*, vol. 25, no. 1, p. 223, Feb. 2025, doi: 10.1186/s12913-024-12162-x.

[41] B. Bajcar, A. Borkowska, and K. Jach, ‘Asymmetry in Usability Evaluation of the Assistive Technology among Users With and Without Disabilities’, *Int. J. Human–Computer Interact.*, vol. 36, no. 19, pp. 1849–1866, Nov. 2020, doi: 10.1080/10447318.2020.1798084.

[42] A. Hanson, H. H. Nap, and the Pillar 1 Workgroup, ‘A scoping review for assistive technologies for healthy ageing: A conversation’, *Gerontechnology*, vol. 23, no. s, pp. 1–1, Jul. 2024, doi: 10.4017/gt.2024.23.s.1107.opp.

[43] WHO, *Global Report on Assistive Technology*, 1st ed. Geneva: World Health Organization, 2022.

[44] K. Morrison, ‘Implementation of assistive computer technology: A model for school systems’, *International Journal of Special Education*, vol. 22, no. 1. pp. 83–95, 2007.

[45] ISO, ‘Assistive products — Classification and terminology’. 2022. [Online]. Available: https://www.iso.org/standard/72464.html

[46] ATIA, ‘What is AT?’, What is AT? [Online]. Available: https://www.atia.org/home/at-resources/what-is-at/

[47] IEEE, ‘Help Build the Future of Assistive Technology’, Empower those in need with a master’s degree in assistive technology engineering. [Online]. Available: https://spectrum.ieee.org/assistive-technology-degree
[48] A. M. Cook and J. M. Polgar, *Assistive technologies: principles and practice*, 4. ed. St. Louis, Missouri: Elsevier/Mosby, 2015.
[49] B. R. Bryant and P. C. Seay, ‘The Technology-Related Assistance to Individuals with Disabilities Act: Relevance to Individuals with Learning Disabilities and Their Advocates’, *J. Learn. Disabil.*, vol. 31, no. 1, pp. 4–15, Jan. 1998, doi: 10.1177/002221949803100102.
[50] WHO and World Bank, ‘World report on disability 2011’, 2011, Accessed: Aug. 20, 2025. [Online]. Available: https://iris.who.int/handle/10665/44575
[51] Organisation mondiale de la santé, Ed., *World health statistics 2016: monitoring health for the SDGs sustainable development goals*. Genève: World health organization, 2016.
[52] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders*, Fifth Edition. American Psychiatric Association, 2013. doi: 10.1176/appi.books.9780890425596.
[53] C. Wohlin, ‘Guidelines for snowballing in systematic literature studies and a replication in software engineering’, in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, London England United Kingdom: ACM, May 2014, pp. 1–10. doi: 10.1145/2601248.2601268.
[54] D. Hernández-Torrano *et al.*, ‘Mental Health and Well-Being of University Students: A Bibliometric Mapping of the Literature’, *Front. Psychol.*, vol. 11, p. 1226, Jun. 2020, doi: 10.3389/fpsyg.2020.01226.
[55] House of Commons, ‘Support for students with mental health issues in higher education in England’. 2019. [Online]. Available: https://dera.ioe.ac.uk/id/eprint/34234/1/CBP-8593.pdf
[56] EPI, ‘Prevalence of mental health issues within the student-aged population’, Children and young people’s mental health. [Online]. Available: https://epi.org.uk/publications-and-research/prevalence-of-mental-health-issues-within-the-student-aged-population/
[57] Í. J. S. Ribeiro, R. Pereira, I. V. Freire, B. G. De Oliveira, C. A. Casotti, and E. N. Boery, ‘Stress and Quality of Life Among University Students: A Systematic Literature Review’, *Health Prof. Educ.*, vol. 4, no. 2, pp. 70–77, Jun. 2018, doi: 10.1016/j.hpe.2017.03.002.
[58] A. Macaskill, ‘The mental health of university students in the United Kingdom’, *Br. J. Guid. Couns.*, vol. 41, no. 4, pp. 426–441, Aug. 2013, doi: 10.1080/03069885.2012.743110.
[59] Catalyst, ‘How to break barriers for women in STEM, tech, and trades’, Women in Science, Technology, Engineering, and Mathematics (STEM) (Quick Take). [Online]. Available: https://www.catalyst.org/research/women-in-science-technology-engineering-and-mathematics-stem/
[60] R. Van Veelen, B. Derks, and M. D. Endedijk, ‘Double Trouble: How Being Outnumbered and Negatively Stereotyped Threatens Career Outcomes of Women in STEM’, *Front. Psychol.*, vol. 10, p. 150, Feb. 2019, doi: 10.3389/fpsyg.2019.00150.
[61] C. Hill, C. Corbett, and A. St. Rose, *Why so few? women in science, technology, engineering, and mathematics*. Washington, D.C: AAUW, 2010.
[62] F. Fajardo Bullón, M. Maestre Campos, E. Felipe Castaño, B. León Del Barco, and M. I. Polo Del Río, ‘ANÁLISIS DEL RENDIMIENTO ACADÉMICO DE LOS ALUMNOS DE EDUCACIÓN SECUNDARIA OBLIGATORIA SEGÚN LAS VARIABLES FAMILIARES’, *Educ. XX1*, vol. 20, no. 1, Oct. 2016, doi: 10.5944/educxx1.17509.
[63] B. Logan and S. Burns, ‘Stressors among young Australian university students: A qualitative study’, *J. Am. Coll. Health*, vol. 71, no. 6, pp. 1753–1760, Jul. 2023, doi: 10.1080/07448481.2021.1947303.
[64] C. S. Hurst, L. E. Baranik, and F. Daniel, ‘College Student Stressors: A Review of the Qualitative Research’, *Stress Health*, vol. 29, no. 4, pp. 275–285, Oct. 2013, doi: 10.1002/smi.2465.
[65] R. Wang *et al.*, ‘StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones’, in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Seattle Washington: ACM, Sep. 2014, pp. 3–14. doi: 10.1145/2632048.2632054.
[66] National institute of mental health, ‘I’M SO STRESSED OUT!’ 2020. [Online]. Available: https://www.nimh.nih.gov/sites/default/files/documents/health/publications/so-stressed-out-fact-sheet/Im-So-Stressed-Out.pdf
[67] H. Jamil, M. Alakkari, M. S. Al-Mahini, M. Alsayid, and O. Al Jandali, ‘The Impact of Anxiety and Depression on Academic Performance: A Cross-Sectional Study among Medical Students in Syria’, *Avicenna J. Med.*, vol. 12, no. 03, pp. 111–119, Jul. 2022, doi: 10.1055/s-0042-1755181.
[68] A. Waqas, A. Rehman, A. Malik, U. Muhammad, S. Khan, and N. Mahmood, ‘Association of Ego Defense Mechanisms with Academic Performance, Anxiety and Depression in Medical Students: A Mixed Methods Study’, *Cureus*, Sep. 2015, doi: 10.7759/cureus.337.

[69] K. H. Teigen, ‘Yerkes-Dodson: A Law for all Seasons’, *Theory Psychol.*, vol. 4, no. 4, pp. 525–547, Nov. 1994, doi: 10.1177/0959354394044004.
[70] B. Andrews and J. M. Wilding, ‘The relation of depression and anxiety to life-stress and achievement in students’, *Br. J. Psychol.*, vol. 95, no. 4, pp. 509–521, Nov. 2004, doi: 10.1348/0007126042369802.
[71] A. Hysenbegasi, S. L. Hass, and C. R. Rowland, ‘The impact of depression on the academic productivity of university students.’, *J. Ment. Health Policy Econ.*, vol. 8, no. 3, pp. 145–151, Sep. 2005.
[72] S. Zada, Y. Wang, M. Zada, and F. Gul, ‘Effect of Mental Health Problems on Academic Performance among University Students in Pakistan’, *Int. J. Ment. Health Promot.*, vol. 23, no. 3, pp. 395–408, 2021, doi: 10.32604/IJMHP.2021.015903.
[73] T. Archer and D. Garcia, ‘Physical Exercise Influences Academic Performance and Well-being in Children and Adolescents’, *Int. J. Sch. Cogn. Psychol.*, vol. 1, no. 1, 2014, doi: 10.4172/2469-9837.1000e102.
[74] H. Raley, J. Naber, S. Cross, and M. Perlow, ‘The Impact of Duration of Sleep on Academic Performance in University Students’, *Madridge J. Nurs.*, vol. 1, no. 1, pp. 11–18, Jul. 2016, doi: 10.18689/mjn-1000103.
[75] M. Hirshkowitz *et al.*, ‘National Sleep Foundation’s updated sleep duration recommendations: final report’, *Sleep Health*, vol. 1, no. 4, pp. 233–243, Dec. 2015, doi: 10.1016/j.sleh.2015.10.004.
[76] R. K. Humphries, D. M. Bath, and N. W. Burton, ‘Dysfunctional beliefs, sleep hygiene and sleep quality in university students’, *Health Promot. J. Austr.*, vol. 33, no. 1, pp. 162–169, Jan. 2022, doi: 10.1002/hpja.471.
[77] T. Kuhn, J. J. Heisz, and L. E. Middleton, ‘The impact of sleep, mental health, and gender on academic performance in Canadian university students’, *Front. Educ.*, vol. 10, p. 1565920, May 2025, doi: 10.3389/feduc.2025.1565920.
[78] D. Smith and R. McLellan, ‘Mental health problems in first-generation university students: A scoping review’, *Rev. Educ.*, vol. 11, no. 3, p. e3418, Dec. 2023, doi: 10.1002/rev3.3418.
[79] M. M. Chemers, L. Hu, and B. F. Garcia, ‘Academic self-efficacy and first year college student performance and adjustment.’, *J. Educ. Psychol.*, vol. 93, no. 1, pp. 55–64, Mar. 2001, doi: 10.1037/0022-0663.93.1.55.
[80] T. Chu, X. Liu, S. Takayanagi, T. Matsushita, and H. Kishimoto, ‘Association between mental health and academic performance among university undergraduates: The interacting role of lifestyle behaviors’, *Int. J. Methods Psychiatr. Res.*, vol. 32, no. 1, p. e1938, Mar. 2023, doi: 10.1002/mpr.1938.
[81] J. Inquilla-Mamani, G. B. Quispe, F. F. Chaiña-Chura, E. Flores-Mamani, J. Apaza-Ticona, and D. J. Incacutipa-Limachi, ‘Impact of Mental Health on Academic Performance: A Study on the Risk of Failure in Higher Education’, *J. Ecohumanism*, vol. 3, no. 7, pp. 4249–4260, Oct. 2024, doi: 10.62754/joe.v3i7.4527.
[82] M. J. Duncan, K. A. Patte, and S. T. Leatherdale, ‘Mental Health Associations with Academic Performance and Education Behaviors in Canadian Secondary School Students’, *Can. J. Sch. Psychol.*, vol. 36, no. 4, pp. 335–357, Dec. 2021, doi: 10.1177/0829573521997311.
[83] Chris Belfield, Jack Britton, Lorraine Dearden, and Laura van der Erv, ‘Higher Education funding in England: past, present and options for the future’, Higher Education funding in England: past, present and options for the future. [Online]. Available: https://ifs.org.uk/publications/higher-education-funding-england-past-present-and-options-future
[84] C. Guan, X. Lu, X. Li, E. Chen, W. Zhou, and H. Xiong, ‘Discovery of College Students in Financial Hardship’, in *2015 IEEE International Conference on Data Mining*, Atlantic City, NJ, USA: IEEE, Nov. 2015, pp. 141–150. doi: 10.1109/ICDM.2015.49.
[85] J. Benson-Egglenton, ‘The financial circumstances associated with high and low wellbeing in undergraduate students: a case study of an English Russell Group institution’, *J. Furth. High. Educ.*, vol. 43, no. 7, pp. 901–913, Aug. 2019, doi: 10.1080/0309877X.2017.1421621.
[86] B. W. Smith, D. Maneze, L. M. Ramjan, C. M. Stephen, J. Montayre, and Y. Salamonson, ‘Impacts of Financial Stress on Mental Health and Wellbeing of Nursing Students: A Systematic Integrative Review’, *J. Adv. Nurs.*, p. jan.70043, Jul. 2025, doi: 10.1111/jan.70043.
[87] R. Humphrey and P. McCarthy, ‘Stress and the Contemporary Student’, *High. Educ. Q.*, vol. 52, no. 2, pp. 221–242, Apr. 1998, doi: 10.1111/1468-2273.00093.
[88] V. K. G. Lim and T. S. H. Teo, ‘Sex, money and financial hardship: An empirical study of attitudes towards money among undergraduates in Singapore’, *J. Econ. Psychol.*, vol. 18, no. 4, pp. 369–386, Jun. 1997, doi: 10.1016/S0167-4870(97)00013-5.
[89] C. Lange and M. Byrd, ‘The Relationship Between Perceptions of Financial Distress and Feelings of Psychological Well-being in New Zealand University Students’, *Int. J. Adolesc. Youth*, vol. 7, no. 3, pp. 193–209, Jan. 1998, doi: 10.1080/02673843.1998.9747824.

[90] J. M. Norvilitis, P. B. Szablicki, and S. D. Wilson, ‘Factors Influencing Levels of Credit-Card Debt in College Students[1]’, *J. Appl. Soc. Psychol.*, vol. 33, no. 5, pp. 935–947, May 2003, doi: 10.1111/j.1559-1816.2003.tb01932.x.

[91] S. Ross, J. Cleland, and M. J. Macleod, ‘Stress, debt and undergraduate medical student performance’, *Med. Educ.*, vol. 40, no. 6, pp. 584–589, Jun. 2006, doi: 10.1111/j.1365-2929.2006.02448.x.

[92] A. Hunt, I. Lincoln, and A. Walker, ‘Term-time employment and academic attainment: evidence from a large-scale survey of undergraduates at Northumbria University’, *J. Furth. High. Educ.*, vol. 28, no. 1, pp. 3–18, Feb. 2004, doi: 10.1080/0309877032000161788.

[93] R. Cooke, M. Barkham, K. Audin, M. Bradley, and J. Davy, ‘Student debt and its relation to student mental health’, *J. Furth. High. Educ.*, vol. 28, no. 1, pp. 53–66, Feb. 2004, doi: 10.1080/0309877032000161814.

[94] R. Roberts, J. Golding, T. Towell, and I. Weinreb, ‘The Effects of Economic Circumstances on British Students’ Mental and Physical Health’, *J. Am. Coll. Health*, vol. 48, no. 3, pp. 103–109, Nov. 1999, doi: 10.1080/07448489909595681.

[95] R. Upsher *et al.*, ‘Understanding how the university curriculum impacts student wellbeing: a qualitative study’, *High. Educ.*, Dec. 2022.

[96] T. Wornast, ‘Exploring student wellbeing in higher education’, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:217417645

[97] S. Brooman, S. Darwent, and A. Pimor, ‘The student voice in higher education curriculum design: is there value in listening?’, *Innov. Educ. Teach. Int.*, vol. 52, no. 6, pp. 663–674, Nov. 2015, doi: 10.1080/14703297.2014.910128.

[98] T. D. Steuber *et al.*, ‘Impact of Curricular Content Reduction on Student-Reported Mental and Physical Health’, *Am. J. Pharm. Educ.*, vol. 89, no. 3, p. 101367, Mar. 2025, doi: 10.1016/j.ajpe.2025.101367.

[99] A.-M. Houghton and J. Anderson, ‘Embedding mental wellbeing in the curriculum: maximising success in higher education’, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:158399119

[100] E. Riva, R. Freeman, L. Schrock, V. Jelicic, C.-T. Ozer, and R. Caleb, ‘Student Wellbeing in the Teaching and Learning Environment: A Study Exploring Student and Staff Perspectives’, *High. Educ. Stud.*, vol. 10, no. 4, p. 103, Nov. 2020, doi: 10.5539/hes.v10n4p103.

[101] E. Jones, M. Priestley, L. Brewster, S. J. Wilbraham, G. Hughes, and L. Spanner, ‘Student wellbeing and assessment in higher education: the balancing act’, *Assess. Eval. High. Educ.*, vol. 46, no. 3, pp. 438–450, Apr. 2021, doi: 10.1080/02602938.2020.1782344.

[102] R. A. Kusurkar *et al.*, ‘The Effect of Assessments on Student Motivation for Learning and Its Outcomes in Health Professions Education: A Review and Realist Synthesis’, *Acad. Med.*, vol. 98, no. 9, pp. 1083–1092, Sep. 2023, doi: 10.1097/ACM.0000000000005263.

[103] K. Piper, J. Morphet, and J. Bonnamy, ‘Improving student-centered feedback through self-assessment’, *Nurse Educ. Today*, vol. 83, p. 104193, Dec. 2019, doi: 10.1016/j.nedt.2019.08.011.

[104] Z. Yan, E. Panadero, X. Wang, and Y. Zhan, ‘A Systematic Review on Students’ Perceptions of Self-Assessment: Usefulness and Factors Influencing Implementation’, *Educ. Psychol. Rev.*, vol. 35, no. 3, p. 81, Sep. 2023, doi: 10.1007/s10648-023-09799-1.

[105] R. Crosby, M. Devlin, and L. Marshall, ‘Computing Ipsative Assessment. Improving the Student Experience in Higher Education Using Personalised Assessment and Feedback’, in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, Aberdeen Scotland Uk: ACM, Jul. 2019, pp. 300–300. doi: 10.1145/3304221.3325565.

[106] H. Jiao, ‘Enhancing students’ engagement in learning through a formative e-assessment tool that motivates students to take action on feedback’, *Australas. J. Eng. Educ.*, vol. 20, no. 1, pp. 9–18, Jan. 2015, doi: 10.7158/D13-002.2015.20.1.

[107] M. N. Demaidi, M. M. Gaber, and N. Filer, ‘OntoPeFeGe: Ontology-Based Personalized Feedback Generator’, *IEEE Access*, vol. 6, pp. 31644–31664, 2018, doi: 10.1109/ACCESS.2018.2846398.

[108] AdvanceHe, ‘Education for Mental Health Toolkit - Workload’, Education for Mental Health Toolkit - Workload. [Online]. Available: https://www.advance-he.ac.uk/teaching-and-learning/curricula-development/education-mental-health-toolkit/learning-focused/workload

[109] A. Navarro, R. F. Armenta, and M. M. Rice, ‘“We are way too stressed”: Balancing Academic Priorities with Mental Health and Wellness Among Life Science Undergraduates’, *CBE—Life Sci. Educ.*, vol. 24, no. 2, p. ar23, Jun. 2025, doi: 10.1187/cbe.24-09-0226.

[110] P. Neigel, A. Vargo, B. Tag, and K. Kise, ‘Identifying Periods of Cyclical Stress in University Students Using Wearables In-the-Wild’, 2024, *arXiv*. doi: 10.48550/ARXIV.2402.11823.

[111] H. J. Park *et al.*, ‘Investigating Cortisol in a STEM Classroom: The Association Between Cortisol and Academic Performance’, *Pers. Soc. Psychol. Bull.*, vol. 51, no. 3, pp. 357–373, Mar. 2025, doi: 10.1177/01461672231188277.

[112] N. Stormon, P. J. Ford, and D. S. Eley, ‘DREEM -ing of dentistry: Students’ perception of the academic learning environment in Australia’, *Eur. J. Dent. Educ.*, vol. 23, no. 1, pp. 35–41, Feb. 2019, doi: 10.1111/eje.12384.

[113] W. Larcombe, C. Baik, and S. Finch, ‘Exploring course experiences that predict psychological distress and mental wellbeing in Australian undergraduate and graduate coursework students’, *High. Educ. Res. Dev.*, vol. 41, no. 2, pp. 420–435, Feb. 2022, doi: 10.1080/07294360.2020.1865284.

[114] Mukta Goyal, Divakar Yadav, and Alka Choubey, ‘E-learning: Current State of Art and Future Prospects’, *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 3, 2012, [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/38584112/25_E-learning_Current_State_of_Art_and_Future_Prospects-libre.pdf?1440649483=&response-content-disposition=inline%3B+filename%3DE_learning_Current_State_of_Art_and_Futu.pdf&Expires=1757804107&Signature=b4HAHip78EFBvsf73zLm3OHrr8UlQRlqPNSXVTjB7T8XLxID--RiN9bBfKllk2ToKrSYRg0Ld~4zS0o3hI2fYGIpTjakWrNWK~Eo0PwO~mIOk35j~OwNsYsSaxQ~zX8GuPv7bMkqrt0pDGz2lR01UvT5y2CyBETbTib5n4gav5U139lv1r4bLXby0nLaTEKXbxLNyhS~O3MCGSROpMJyi8MN5RdjgWFOM9NpgLJAFkmM1bCxP6dPJ1pb-lAQKXQTHZa~zhuO3K~e24nBQ41zlZ5LTFBikAF4R2tTlvkjnqEjgzYNgaz0vTHRcmmhr0nPoDmtxGLQGQ~NJSCDtxjoqw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[115] K. E. Rakow, R. J. Upsher, J. L. H. Foster, N. C. Byrom, and E. J. Dommett, ‘“It Ain’t What You Use, It’s the Way That You Use It”: How Virtual Learning Environments May Impact Student Mental Wellbeing’, *Educ. Sci.*, vol. 13, no. 7, p. 749, Jul. 2023, doi: 10.3390/educsci13070749.

[116] M. Shkoukani, ‘Explore the Major Characteristics of Learning Management Systems and their Impact on e-Learning Success’, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, 2019, doi: 10.14569/IJACSA.2019.0100139.

[117] S. H. P. W. Gamage, J. R. Ayres, and M. B. Behrend, ‘A systematic review on trends in using Moodle for teaching and learning’, *Int. J. STEM Educ.*, vol. 9, no. 1, p. 9, Dec. 2022, doi: 10.1186/s40594-021-00323-x.

[118] *WHO Guidelines on Mental Health at Work*, 1st ed. Geneva: World Health Organization, 2022.

[119] Elizabeth Hampson, AAnastasiia Polner, Valentine Assal, and Natasha Abrahams, ‘Mental health and employers’, Deloitte, 2022. [Online]. Available: https://www.deloitte.com/uk/en/services/consulting/analysis/mental-health-and-employers-the-case-for-investment.html

[120] I. Houtman, L. Cedillo, and K. Jettinghoff, *Raising awareness of stress at work in developing countries: a modern hazard in a traditional working environment : advice to employer and worker representatives*. Geneva: World Health Organization, 2007.

[121] A. Marchand, P. Durand, V. Haines, and S. Harvey, ‘The multilevel determinants of workers’ mental health: results from the SALVEO study’, *Soc. Psychiatry Psychiatr. Epidemiol.*, vol. 50, no. 3, pp. 445–459, Mar. 2015, doi: 10.1007/s00127-014-0932-y.

[122] S. B. Harvey *et al.*, ‘Can work make you mentally ill? A systematic meta-review of work-related risk factors for common mental health problems’, *Occup. Environ. Med.*, vol. 74, no. 4, pp. 301–310, Apr. 2017, doi: 10.1136/oemed-2016-104015.

[123] W. J. Hwang and H. H. Jo, ‘Impact of Mental Health on Wellness in Adult Workers’, *Front. Public Health*, vol. 9, p. 743344, Dec. 2021, doi: 10.3389/fpubh.2021.743344.

[124] Richard Wynne *et al.*, ‘Promoting mental health in the workplace’, European Comission, 2014. [Online]. Available: https://ec.europa.eu/social/BlobServlet?docId=13879&langId=en

[125] A. Jain, M. Agarwal, and V. Shubha Rao, ‘Employee Burnout Prediction: A Supervised Learning Approach’, in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, Bangalore, India: IEEE, Oct. 2021, pp. 1–5. doi: 10.1109/GCAT52182.2021.9587830.

[126] I. Hungerbuehler, K. Daley, K. Cavanagh, H. Garcia Claro, and M. Kapps, ‘Chatbot-Based Assessment of Employees’ Mental Health: Design Process and Pilot Implementation’, *JMIR Form. Res.*, vol. 5, no. 4, p. e21678, Apr. 2021, doi: 10.2196/21678.

[127] C. Richard *et al.*, ‘Disclosure Impact of Mental Health Conditions in the Workplace: A Scoping Review and a Thematic Analysis’, *J. Occup. Rehabil.*, Apr. 2025, doi: 10.1007/s10926-025-10288-1.

[128] P. Gray, S. Senabe, N. Naicker, S. Kgalamono, A. Yassi, and J. M. Spiegel, ‘Workplace-Based Organizational Interventions Promoting Mental Health and Happiness among Healthcare Workers: A Realist Review’, *Int. J. Environ. Res. Public. Health*, vol. 16, no. 22, p. 4396, Nov. 2019, doi: 10.3390/ijerph16224396.

[129] W. M. A. Wan Mohd Yunus, P. Musiat, and J. S. L. Brown, ‘Systematic review of universal and targeted workplace interventions for depression’, *Occup. Environ. Med.*, vol. 75, no. 1, pp. 66–75, Jan. 2018, doi: 10.1136/oemed-2017-104532.

[130] J. Joyce, ‘Facing the challenge of mental ill health in the workplace’, *J. Public Ment. Health*, vol. 12, no. 2, pp. 93–97, Jun. 2013, doi: 10.1108/JPMH-01-2013-0003.

[131] A. Wu, E. C. Roemer, K. B. Kent, D. W. Ballard, and R. Z. Goetzel, ‘Organizational Best Practices Supporting Mental Health in the Workplace’, *J. Occup. Environ. Med.*, vol. 63, no. 12, pp. e925–e931, Dec. 2021, doi: 10.1097/JOM.0000000000002407.

[132] A. D. LaMontagne *et al.*, ‘Workplace mental health: developing an integrated intervention approach’, *BMC Psychiatry*, vol. 14, no. 1, p. 131, Dec. 2014, doi: 10.1186/1471-244X-14-131.

[133] H. M. Abdul Fattah, K. M. Azharul Hasan, and S. Das, ‘A Voting Classifier for the Treatment of Employees’ Mental Health Disorder’, in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, Rajshahi, Bangladesh: IEEE, Jul. 2021, pp. 1–6. doi: 10.1109/ACMI53878.2021.9528102.

[134] U. S. Reddy, A. V. Thota, and A. Dharun, ‘Machine Learning Techniques for Stress Prediction in Working Employees’, in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India: IEEE, Dec. 2018, pp. 1–4. doi: 10.1109/ICCIC.2018.8782395.

[135] R. Katarya and S. Maan, ‘Predicting Mental health disorders using Machine Learning for employees in technical and non-technical companies’, in *2020 IEEE International Conference on Advances and Developments in Electrical and Electronics Engineering (ICADEE)*, Coimbatore, India: IEEE, Dec. 2020, pp. 1–5. doi: 10.1109/ICADEE51157.2020.9368923.

[136] L. Mohan and G. Panuganti, ‘Perceived Stress Prediction among Employees using Machine Learning techniques’, in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India: IEEE, Mar. 2022, pp. 1–6. doi: 10.1109/IC3IOT53935.2022.9768026.

[137] M. S. De Lorenzo, ‘Employee Mental Illness: Managing the Hidden Epidemic’, *Empl. Responsib. Rights J.*, vol. 25, no. 4, pp. 219–238, Dec. 2013, doi: 10.1007/s10672-013-9226-x.

[138] E. K. Kelloway, J. K. Dimoff, and S. Gilbert, ‘Mental Health in the Workplace’, *Annu. Rev. Organ. Psychol. Organ. Behav.*, vol. 10, no. 1, pp. 363–387, Jan. 2023, doi: 10.1146/annurev-orgpsych-120920-050527.

[139] Alice Sinclair and Siobhan O’Regan, ‘Mental Health and Work’, INSTITUTE FOR EMPLOYMENT STUDIES, MP75, 2007. [Online]. Available: https://www.employment-studies.co.uk/resource/mental-health-and-work

[140] J. K. Harter, F. L. Schmidt, and C. L. M. Keyes, ‘Well-being in the workplace and its relationship to business outcomes: A review of the Gallup studies.’, in *Flourishing: Positive psychology and the life well-lived.*, C. L. M. Keyes and J. Haidt, Eds, Washington: American Psychological Association, 2003, pp. 205–224. doi: 10.1037/10594-009.

[141] T. A. Wright, ‘To Be Or Not To Be [Happy]: The Role of Employee Well-Being’, *Acad. Manag. Perspect.*, vol. 20, no. 3, pp. 118–120, Aug. 2006, doi: 10.5465/amp.2006.21903486.

[142] T. Rajgopal, ‘Mental well-being at the workplace’, *Indian J. Occup. Environ. Med.*, vol. 14, no. 3, p. 63, 2010, doi: 10.4103/0019-5278.75691.

[143] C. Peng, ‘The Academic Motivation and Engagement of Students in English as a Foreign Language Classes: Does Teacher Praise Matter?’, *Front. Psychol.*, vol. 12, p. 778174, Oct. 2021, doi: 10.3389/fpsyg.2021.778174.

[144] B. Masluk, A. Asensio-Martínez, S. Lopes, and A. Vega-Muñoz, ‘Editorial: Technostress, mental health and well-being’, *Front. Psychol.*, vol. 14, p. 1356844, Jan. 2024, doi: 10.3389/fpsyg.2023.1356844.

[145] H. Hu and H. Luo, ‘Academic motivation among senior students majoring in rehabilitation related professions in China’, *BMC Med. Educ.*, vol. 21, no. 1, p. 582, Dec. 2021, doi: 10.1186/s12909-021-03016-9.

[146] P. Upadhyaya and Vrinda, ‘Impact of technostress on academic productivity of university students’, *Educ. Inf. Technol.*, vol. 26, no. 2, pp. 1647–1664, Mar. 2021, doi: 10.1007/s10639-020-10319-9.

[147] S. Sharma and B. Gupta, ‘Investigating the role of technostress, cognitive appraisal and coping strategies on students’ learning performance in higher education: a multidimensional transactional theory of stress approach’, *Inf. Technol. People*, vol. 36, no. 2, pp. 626–660, Mar. 2023, doi: 10.1108/ITP-06-2021-0505.

[148] J. Mangundu, ‘The effects of technostress on academic commitment among first-year undergraduate students at an institution of higher education in South Africa’, in *Proceedings of the 4th African Human Computer Interaction Conference*, East London South Africa: ACM, Nov. 2023, pp. 106–117. doi: 10.1145/3628096.3629044.

[149] D. R. Perera, ‘Does Technostress Impact On University Students’ Academic Performance in the New Normal?’, *Int. J. Eng. Manag. Res.*, vol. 11, no. 6, Dec. 2021, doi: 10.31033/ijemr.11.6.12.

[150] H. B. Essel, D. Vlachopoulos, A. Tachie-Menson, E. E. Johnson, and A. K. Ebeheakey, ‘Technology-Induced Stress, Sociodemographic Factors, and Association with Academic Achievement and

Productivity in Ghanaian Higher Education during the COVID-19 Pandemic', *Information*, vol. 12, no. 12, p. 497, Nov. 2021, doi: 10.3390/info12120497.
[151] S. Mehrolia, S. Alagarsamy, and Jeevananda S. (8Bcd2Ac9-364C-41Ed-8A8A-249Ca119D7F9, 'The Dark Side of Technology-Enabled Teaching: Impact of Technostress on Student Performance', *Int. J. Mob. Blended Learn.*, vol. 13, no. 3, pp. 1–22, Jul. 2021, doi: 10.4018/IJMBL.2021070103.
[152] J. Galvin *et al.*, 'Technostress, coping, and anxious and depressive symptomatology in university students during the Covid-19 pandemic', *Eur. J. Psychol.*, vol. 18, no. 3, pp. 302–318, Aug. 2022, doi: 10.5964/ejop.4725.
[153] 'ANALYSIS OF TECHNOSTRESS EXPERIENCED BY STUDENTS AT THE UNIVERSITY OF CAPE TOWN, DURING THE COVID-19 PANDEMIC', *Issues Inf. Syst.*, 2022, doi: 10.48009/1_iis_2022_106.
[154] A. Ioannou, M. Lycett, and A. Marshan, 'The Role of Mindfulness in Mitigating the Negative Consequences of Technostress', *Inf. Syst. Front.*, vol. 26, no. 2, pp. 523–549, Apr. 2024, doi: 10.1007/s10796-021-10239-0.
[155] C. Qi, 'A double-edged sword? Exploring the impact of students' academic usage of mobile devices on technostress and academic performance', *Behav. Inf. Technol.*, vol. 38, no. 12, pp. 1337–1354, Dec. 2019, doi: 10.1080/0144929X.2019.1585476.
[156] G. Zhao, Q. Wang, L. Wu, and Y. Dong, 'Exploring the Structural Relationship Between University Support, Students' Technostress, and Burnout in Technology-enhanced Learning', *Asia-Pac. Educ. Res.*, vol. 31, no. 4, pp. 463–473, Aug. 2022, doi: 10.1007/s40299-021-00588-4.
[157] X. Wang, S. C. Tan, and L. Li, 'Technostress in university students' technology-enhanced learning: An investigation from multidimensional person-environment misfit', *Comput. Hum. Behav.*, vol. 105, p. 106208, Apr. 2020, doi: 10.1016/j.chb.2019.106208.
[158] M. Salanova, W. Schaufeli, I. Martínez, and E. Bresó, 'How obstacles and facilitators predict academic performance: the mediating role of study burnout and engagement', *Anxiety Stress Coping*, vol. 23, no. 1, pp. 53–70, Jan. 2010, doi: 10.1080/10615800802609965.
[159] X. Wang, Z. Li, Z. Ouyang, and Y. Xu, 'The Achilles Heel of Technology: How Does Technostress Affect University Students' Wellbeing and Technology-Enhanced Learning', *Int. J. Environ. Res. Public. Health*, vol. 18, no. 23, p. 12322, Nov. 2021, doi: 10.3390/ijerph182312322.
[160] F. Saleem, E. Chikhaoui, and M. I. Malik, 'Technostress in students and quality of online learning: role of instructor and university support', *Front. Educ.*, vol. 9, p. 1309642, Feb. 2024, doi: 10.3389/feduc.2024.1309642.
[161] Q. Wang, G. Zhao, and J. Zeng, 'The effects of facilitating conditions, digital competence, and technostress on higher education students' digital informal learning: A moderated mediation examination', *Australas. J. Educ. Technol.*, Sep. 2024, doi: 10.14742/ajet.9324.
[162] N. N. Abd Aziz, M. A. R. Awang Kader, and R. Ab Halim, 'The Impact of Technostress on Student Satisfaction and Performance Expectancy', *Asian J. Univ. Educ.*, vol. 17, no. 4, p. 538, Dec. 2021, doi: 10.24191/ajue.v17i4.16466.
[163] N. N. Abd Aziz, M. A. Aziz, and N. A. S. Abd Rahman, 'The mediating effects of student satisfaction on technostress–performance expectancy relationship in university students', *J. Appl. Res. High. Educ.*, vol. 15, no. 1, pp. 113–129, Jan. 2023, doi: 10.1108/JARHE-03-2021-0117.
[164] A. Alvarez-Risco, S. Del-Aguila-Arcentales, J. A. Yáñez, M. A. Rosen, and C. R. Mejia, 'Influence of Technostress on Academic Performance of University Medicine Students in Peru during the COVID-19 Pandemic', *Sustainability*, vol. 13, no. 16, p. 8949, Aug. 2021, doi: 10.3390/su13168949.
[165] Óscar. R. González-López, M. Buenadicha-Mateos, and M. I. Sánchez-Hernández, 'Overwhelmed by Technostress? Sensitive Archetypes and Effects in Times of Forced Digitalization', *Int. J. Environ. Res. Public. Health*, vol. 18, no. 8, p. 4216, Apr. 2021, doi: 10.3390/ijerph18084216.
[166] E. A. Hapsari, E. N. Rohmatullayaly, and K. A. Widayati, 'Technostress and Sleep Quality among University Students in Indonesia: A Cross-sectional Study', *Asian J. Soc. Health Behav.*, vol. 7, no. 4, pp. 197–202, Oct. 2024, doi: 10.4103/shb.shb_177_24.
[167] Prince Addai, Michelle Afrifah, Isaac Okyere, Gloria Tachie-Donkor, and Jacob Owusu Sarfo, 'Technostress and Determinants of Academic Success among University Students: Mediation Role of Technological Literacy', *J. Advocacy Res. Educ.*, vol. 11, no. 1, Apr. 2024, doi: 10.13187/jare.2024.1.47.
[168] Y. Y. Choqque-Cabrera, M. J. Malaga-Villalba, and F. M. Talavera-Mendoza, 'Technostress in The Frequency of Use of ICT', in *2021 3rd International Conference on Modern Educational Technology*, Jakarta Indonesia: ACM, May 2021, pp. 93–99. doi: 10.1145/3468978.3468994.
[169] E. Rohwer, J.-C. Flöther, V. Harth, and S. Mache, 'Overcoming the "Dark Side" of Technology—A Scoping Review on Preventing and Coping with Work-Related Technostress', *Int. J. Environ. Res. Public. Health*, vol. 19, no. 6, p. 3625, Mar. 2022, doi: 10.3390/ijerph19063625.

[170] A. M. Fuglseth and Ø. Sørebø, 'The effects of technostress within the context of employee use of ICT', *Comput. Hum. Behav.*, vol. 40, pp. 161–170, 2014, doi: https://doi.org/10.1016/j.chb.2014.07.040.

[171] Henner Gimpel, Julia Lanzl, Tobias Manner-Romberg, and Niclas Nuske, 'A survey of employees on the stress and strain caused by working with digital technologies', Hans Böckler Stiftung, 2018. [Online]. Available: https://www.boeckler.de/de/faust-detail.htm?sync_id=HBS-007024

[172] R. Oksa, T. Saari, M. Kaakinen, and A. Oksanen, 'The Motivations for and Well-Being Implications of Social Media Use at Work among Millennials and Members of Former Generations', *Int. J. Environ. Res. Public. Health*, vol. 18, no. 2, p. 803, Jan. 2021, doi: 10.3390/ijerph18020803.

[173] T. E. Rizzuto, 'Age and technology innovation in the workplace: Does work context matter?', *Comput. Hum. Behav.*, vol. 27, no. 5, pp. 1612–1620, Sep. 2011, doi: 10.1016/j.chb.2011.01.011.

[174] A. Raza, M. I. Ishaq, H. Zia, Z. ur-Rehman, and R. Ahmad, 'Technostressors and service employees outcomes: a longitudinal study', *Serv. Ind. J.*, vol. 42, no. 13–14, pp. 1030–1053, Oct. 2022, doi: 10.1080/02642069.2022.2081685.

[175] Adamkolo Mohammed Ibrahim, Mohd Nizam Osman, Ahmed Lawal Gusau, and Phuong Thi VI, 'Correlation of Technostress Creators with Employees' Work-Life Balance in the Context of Journalists' Use of Information and Communication Technology at Work: Moderating Role of Self-Efficacy', *Int. J. Media Inf. Lit.*, vol. 6, no. 2, Dec. 2021, doi: 10.13187/ijmil.2021.2.338.

[176] H. Wang, H. Ding, and X. Kong, 'Understanding technostress and employee well-being in digital work: the roles of work exhaustion and workplace knowledge diversity', *Int. J. Manpow.*, vol. 44, no. 2, pp. 334–353, Apr. 2023, doi: 10.1108/IJM-08-2021-0480.

[177] M. Kotek and I. Vranjes, 'When do Employees Perceive Technology as Stressful? A Meta-Analysis of work-related Technostress Antecedents', *Inf. Manage.*, vol. 62, no. 8, p. 104203, Dec. 2025, doi: 10.1016/j.im.2025.104203.

[178] R. J. P. Curbano, 'Mediating Effects of Technostress on the Interplay of Workplace Design and Physical Discomfort Among Employees of Business Outsourcing Industries', in *2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Bangkok, Thailand: IEEE, Dec. 2024, pp. 1054–1058. doi: 10.1109/IEEM62345.2024.10857178.

[179] D. M. Marchiori, E. W. Mainardes, and R. G. Rodrigues, 'Do Individual Characteristics Influence the Types of Technostress Reported by Workers?', *Int. J. Human–Computer Interact.*, vol. 35, no. 3, pp. 218–230, Feb. 2019, doi: 10.1080/10447318.2018.1449713.

[180] M. Harunavamwe and C. Ward, 'The influence of technostress, work–family conflict, and perceived organisational support on workplace flourishing amidst COVID-19', *Front. Psychol.*, vol. 13, p. 921211, Jul. 2022, doi: 10.3389/fpsyg.2022.921211.

[181] P. Kot, 'Role of Technostress in Job Satisfaction and Work Engagement in People Working with Information and Communication Technologies', *Pak. J. Psychol. Res.*, vol. 37, no. 3, pp. 331–349, Jul. 2022, doi: 10.33824/PJPR.2022.37.3.20.

[182] A. Ioannou, 'Mindfulness and technostress in the workplace: a qualitative approach', *Front. Psychol.*, vol. 14, p. 1252187, Nov. 2023, doi: 10.3389/fpsyg.2023.1252187.

[183] K. Pflügner, C. Maier, and T. Weitzel, 'The direct and indirect influence of mindfulness on techno-stressors and job burnout: A quantitative study of white-collar workers', *Comput. Hum. Behav.*, vol. 115, p. 106566, Feb. 2021, doi: 10.1016/j.chb.2020.106566.

[184] I. Vranjes, A. Lacroix, and A. Van Den Broeck, 'Developing a MODI Framework for Motivational Organizational Disconnection (MODI) Policies', in *Humanizing the Digital Workspace*, M. Orel, M. Černe, and S. I. Wong, Eds, Cham: Springer Nature Switzerland, 2025, pp. 359–382. doi: 10.1007/978-3-031-76902-3_15.

[185] N. Vendramin, G. Nardelli, and C. Ipsen, 'Task-Technology Fit Theory', in *A Handbook of Theories on Designing Alignment between People and the Office Environment*, 1st edn, London: Routledge, 2021, pp. 39–53. doi: 10.1201/9781003128830-4.

[186] M. Salah-Eddine, M. El Hamlaoui, and M. Belaissaoui, 'Computerised method of coping with technostress in organizational situation', in *2018 International Conference on Information Management and Processing (ICIMP)*, London: IEEE, Jan. 2018, pp. 130–134. doi: 10.1109/ICIMP1.2018.8325854.

[187] M. Salah-Eddine and M. Belaissaoui, 'Technostress, Coping and Job Satisfaction Model of Information Systems', in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA: IEEE, Dec. 2016, pp. 139–142. doi: 10.1109/CSCI.2016.0033.

[188] K. Wang, Q. Shu, and Q. Tu, 'Empirical study of coping strategies for computer-related technostress of Chinese employees', *Tsinghua Sci. Technol.*, vol. 10, pp. 753–760, Dec. 2005.

[189] L. Carvajal-Velez *et al.*, 'Increasing Data and Understanding of Adolescent Mental Health Worldwide: UNICEF's Measurement of Mental Health Among Adolescents at the Population Level Initiative', *J. Adolesc. Health*, vol. 72, no. 1, pp. S12–S14, Jan. 2023, doi: 10.1016/j.jadohealth.2021.03.019.

[190] M. O. Ogundele and M. Morton, 'Classification, prevalence and integrated care for neurodevelopmental and child mental health disorders: A brief overview for paediatricians', *World J. Clin. Pediatr.*, vol. 11, no. 2, pp. 120–135, Mar. 2022, doi: 10.5409/wjcp.v11.i2.120.

[191] C. R. Bugaj, *The new assistive tech: make learning awesome for all!*, Second edition. Arlington, VA: International Society for Technology in Education, 2023.

[192] G. Yenduri *et al.*, 'From Assistive Technologies to Metaverse—Technologies in Inclusive Higher Education for Students With Specific Learning Difficulties: A Review', *IEEE Access*, vol. 11, pp. 64907–64927, 2023, doi: 10.1109/ACCESS.2023.3289496.

[193] J. M. Fernández-Batanero, M. Montenegro-Rueda, J. Fernández-Cerero, and I. García-Martínez, 'Assistive technology for the inclusion of students with disabilities: a systematic review', *Educ. Technol. Res. Dev.*, vol. 70, no. 5, pp. 1911–1930, Oct. 2022, doi: 10.1007/s11423-022-10127-7.

[194] M. M. Abdelwahab, K. A. Al-Karawi, and H. E. Semary, 'A Systematic Review of Assistive Technology for Enhancing the Students with Disabilities', *J. Disabil. Res.*, vol. 4, no. 2, 2025, doi: 10.57197/JDR-2024-0117.

[195] S. E. Kisanga and D. H. Kisanga, 'The role of assistive technology devices in fostering the participation and learning of students with visual impairment in higher education institutions in Tanzania', *Disabil. Rehabil. Assist. Technol.*, vol. 17, no. 7, pp. 791–800, Oct. 2022, doi: 10.1080/17483107.2020.1817989.

[196] D. Chambers, 'Assistive Technology Supporting Inclusive Education: Existing and Emerging Trends', in *International Perspectives on Inclusive Education*, D. Chambers, Ed., Emerald Publishing Limited, 2020, pp. 1–16. doi: 10.1108/S1479-363620200000014001.

[197] M. P. Malcolm and M. C. Roll, 'The impact of assistive technology services in post-secondary education for students with disabilities: Intervention outcomes, use-profiles, and user-experiences', *Assist. Technol.*, vol. 29, no. 2, pp. 91–98, Apr. 2017, doi: 10.1080/10400435.2016.1214932.

[198] M. N. Lartz, J. B. Stoner, and L. Stout, 'Perspectives of Assistive Technology from Deaf Students at a Hearing University.', 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:73731123

[199] A. McNicholl, D. Desmond, and P. Gallagher, 'Assistive technologies, educational engagement and psychosocial outcomes among students with disabilities in higher education', *Disabil. Rehabil. Assist. Technol.*, vol. 18, no. 1, pp. 50–58, Jan. 2023, doi: 10.1080/17483107.2020.1854874.

[200] A. McNicholl, H. Casey, D. Desmond, and P. Gallagher, 'The impact of assistive technology use for students with disabilities in higher education: a systematic review', *Disabil. Rehabil. Assist. Technol.*, vol. 16, no. 2, pp. 130–143, Feb. 2021, doi: 10.1080/17483107.2019.1642395.

[201] G. Ketema Dabi and D. Negassa Golga, 'The role of assistive technology in supporting the engagement of students with visual impairment in learning mathematics: An integrative literature review', *Br. J. Vis. Impair.*, vol. 42, no. 3, pp. 674–687, Sep. 2024, doi: 10.1177/02646196231158922.

[202] K. P. Vinumol, A. Chowdhury, R. Kambam, and V. Muralidharan, 'Augmented Reality Based Interactive Text Book: An Assistive Technology for Students with Learning Disability', in *2013 XV Symposium on Virtual and Augmented Reality*, Cuiabá - Mato Grosso, Brazil: IEEE, May 2013, pp. 232–235. doi: 10.1109/SVR.2013.26.

[203] J. A. Black and D. S. Hayden, 'The Note-Taker: An assistive technology that allows students who are legally blind to take notes in the classroom', in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, San Francisco, CA: IEEE, Jun. 2010, pp. 1–8. doi: 10.1109/CVPRW.2010.5543587.

[204] Alsoori Alzaabi, Abdulla, 'Assistive Technology's Effect on the Academic Performance of Students with Disabilities: An Investigative Study at Higher Education Institutions in the UAE', The British University in Dubai (BUiD), 2020. [Online]. Available: https://bspace.buid.ac.ae/handle/1234/1765

[205] M. P. Malcolm and M. C. Roll, 'Assistive technology outcomes in post-secondary students with disabilities: the influence of diagnosis, gender, and class-level', *Disabil. Rehabil. Assist. Technol.*, vol. 12, no. 8, pp. 857–867, Nov. 2017, doi: 10.1080/17483107.2016.1277794.

[206] P. Reddy, E. Reddy, V. Chand, S. Paea, and A. Prasad, 'Assistive Technologies: Saviour of Mathematics in Higher Education', *Front. Appl. Math. Stat.*, vol. 6, p. 619725, Feb. 2021, doi: 10.3389/fams.2020.619725.

[207] E. Radwa, A. Fahad, F. Shereen, A. Fahad, M. Yasser, and E. Mohamed, 'Bridging to possible: Empowering visually impaired students through assistive technology rehabilitative training to become better students and future employees', *Edelweiss Appl. Sci. Technol.*, vol. 9, no. 4, pp. 1160–1175, Apr. 2025, doi: 10.55214/25768484.v9i4.6190.

[208] O. Poobrasert and W. Gestubtim, 'Development of assistive technology for students with dyscalculia', in *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)*, Lodz, Poland: IEEE, Sep. 2013, pp. 60–63. doi: 10.1109/ICeLeTE.2013.6644348.

[209] O. Poobrasert, P. Sabayjai, and N. Mitsamarn, 'Technology-enhanced Learning for Students with Learning Disabilities', in *2011 IEEE 11th International Conference on Advanced Learning Technologies*, Athens, GA, USA: IEEE, Jul. 2011, pp. 496–497. doi: 10.1109/ICALT.2011.154.
[210] House of Commons, 'Assistive technology', HC 673 10, 2018. [Online]. Available: https://publications.parliament.uk/pa/cm201719/cmselect/cmworpen/673/673.pdf
[211] M. Linden and K. Milchus, 'Teleworkers with disabilities: Characteristics and accommodation use', *Work*, vol. 47, no. 4, pp. 473–483, 2014, doi: 10.3233/WOR-141834.
[212] V. Kalargyrou, 'Gaining a Competitive Advantage with Disability Inclusion Initiatives', *J. Hum. Resour. Hosp. Tour.*, vol. 13, no. 2, pp. 120–145, Apr. 2014, doi: 10.1080/15332845.2014.847300.
[213] Neha Kumari and Usha Lenka, 'Employment and Retention of Differently-abled People in the Workplace Through Assistive Technologies', *Int. J. Digit. Technol.*, vol. 2, no. 1, 2023, [Online]. Available: https://journal.nielit.edu.in/index.php/01/article/view/28
[214] F. Sacchi, 'Promoting Labour Market Inclusion: a Review of Assistive Technologies for Workers with Autism Spectrum Disorder', *ICCHP-AAATE 2022 Open Access Compend. Assist. Technol.*, vol. Accessibility and, p. 8 pages, 2022, doi: 10.35011/ICCHP-AAATE22-P1-13.
[215] E. Beneteau, H. Feldner, and W. Pratt, '"I miss work:" employment experiences and attitudes of adults with acquired disabilities who use assistive technologies', *Disabil. Rehabil. Assist. Technol.*, vol. 19, no. 4, pp. 1600–1609, May 2024, doi: 10.1080/17483107.2023.2221312.
[216] B. Phillips and H. Zhao, 'Predictors of Assistive Technology Abandonment', *Assist. Technol.*, vol. 5, no. 1, pp. 36–45, Jun. 1993, doi: 10.1080/10400435.1993.10132205.
[217] M. Jakovljevic and S. Buckley, 'Assistive Technologies in a Workplace Environment: Barriers for the Employment of People with Disabilities', *Disabil. CBR Incl. Dev.*, vol. 22, no. 2, p. 55, Oct. 2011, doi: 10.5463/dcid.v22i2.32.
[218] D. M. De Jonge and S. A. Rodger, 'Consumer-identified barriers and strategies for optimizing technology use in the workplace', *Disabil. Rehabil. Assist. Technol.*, vol. 1, no. 1–2, pp. 79–88, Jan. 2006, doi: 10.1080/09638280500167324.
[219] S. M. Branham and S. K. Kane, 'The Invisible Work of Accessibility: How Blind Employees Manage Accessibility in Mixed-Ability Workplaces', in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*, Lisbon, Portugal: ACM Press, 2015, pp. 163–171. doi: 10.1145/2700648.2809864.
[220] K. Jääskeläinen and N. Nevala, 'Use of Assistive Technology in Workplaces of Employees with Physical and Cognitive Disabilities', in *Computers Helping People with Special Needs*, vol. 7382, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds, in Lecture Notes in Computer Science, vol. 7382. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 223–226. doi: 10.1007/978-3-642-31522-0_33.
[221] O. Aleksandrova and Y. Nenakhova, 'Accessibility of Assistive Technologies as a Factor in the Successful Realization of the Labor Potential of Persons with Disabilities: Russia's Experience', *Societies*, vol. 9, no. 4, p. 70, Oct. 2019, doi: 10.3390/soc9040070.
[222] K. Shinohara and J. O. Wobbrock, 'In the shadow of misperception: assistive technology use and social interactions', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver BC Canada: ACM, May 2011, pp. 705–714. doi: 10.1145/1978942.1979044.
[223] M. Jakovljevic and S. Buckley, 'Assistive Technologies in a Workplace Environment: Barriers for the Employment of People with Disabilities', *Disabil. CBR Incl. Dev.*, vol. 22, no. 2, p. 55, Oct. 2011, doi: 10.5463/dcid.v22i2.32.
[224] A. Reinersmann, M. Heinz-Jakobs, E. Bicker, and C. Röcker, 'Examples of Implementing Digital Assistive Technologies in the Open Labor Market', in *HCI International 2025 Posters*, vol. 2524, C. Stephanidis, M. Antona, S. Ntoa, and G. Salvendy, Eds, in Communications in Computer and Information Science, vol. 2524. , Cham: Springer Nature Switzerland, 2025, pp. 161–171. doi: 10.1007/978-3-031-94156-6_17.
[225] S. Lackey *et al.*, 'Barriers and facilitators to accommodations in the workplace for adults who use augmentative and alternative communication (AAC): a systematic review', *Augment. Altern. Commun.*, vol. 39, no. 3, pp. 181–197, Jul. 2023, doi: 10.1080/07434618.2023.2170277.
[226] J. Wong *et al.*, 'Job Accommodations, Return to Work and Job Retention of People with Physical Disabilities: A Systematic Review', *J. Occup. Rehabil.*, vol. 31, no. 3, pp. 474–490, Sep. 2021, doi: 10.1007/s10926-020-09954-3.
[227] D. D. Jonge, S. Rodger, and H. Fitzgibbon, 'Putting technology to work: Users' perspective on integrating assistive technology into the workplace', *WORK J. Prev. Assess. Rehabil.*, vol. 16, no. 2, pp. 77–89, Jan. 2001, doi: 10.3233/WOR-2001-00139.

[228] A. J. Langton and H. Ramseur, 'Enhancing employment outcomes through job accommodation and assistive technology resources and services', *J. Vocat. Rehabil.*, vol. 16, no. 1, pp. 27–37, Jan. 2001, doi: 10.3233/JVR-2001-00104.

[229] S. Arthanat, K. Lesner, and V. Sundar, 'An evaluation framework to measure usability of Assistive Technology at workplace: A demonstration study', *J. Vocat. Rehabil.*, vol. 44, no. 2, pp. 213–226, Mar. 2016, doi: 10.3233/JVR-150792.

[230] A. L. Sauer, A. Parks, and P. C. Heyn, 'Assistive technology effects on the employment outcomes for people with cognitive disabilities: A systematic review', *Disabil. Rehabil. Assist. Technol.*, vol. 5, no. 6, pp. 377–391, Nov. 2010, doi: 10.3109/17483101003746360.

[231] W. Strobel and J. T. McDonough, 'Workplace personal assistance service and assistive technology', *J. Vocat. Rehabil.*, vol. 18, no. 2, pp. 107–112, Mar. 2003, doi: 10.3233/JVR-2003-00186.

[232] D. Kruse *et al.*, 'Assistive Technology's Potential to Improve Employment of People with Disabilities', *J. Occup. Rehabil.*, vol. 34, no. 2, pp. 299–315, Jun. 2024, doi: 10.1007/s10926-023-10164-w.

[233] R. A. Rahmatika, C. P. Pratiwi, and C. Basuki, 'Does the Provision of Assistive Technology Increase Disability Employment?', *IJDS Indones. J. Disabil. Stud.*, vol. 9, no. 02, pp. 179–201, Dec. 2022, doi: 10.21776/ub.ijds.2022.009.02.04.

[234] S. Reeder, L. Kelly, B. Kechavarzi, and S. Sabanovic, 'Breakbot: a social motivator for the workplace', in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, Aarhus Denmark: ACM, Aug. 2010, pp. 61–64. doi: 10.1145/1858171.1858184.

[235] M. Funk and A. Schmidt, 'Cognitive Assistance in the Workplace', *IEEE Pervasive Comput.*, vol. 14, no. 3, pp. 53–55, Jul. 2015, doi: 10.1109/MPRV.2015.53.

[236] R. Scherer, F. Siddiq, and B. Sánchez Viveros, 'A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions', *Comput. Hum. Behav.*, vol. 109, p. 106349, Aug. 2020, doi: 10.1016/j.chb.2020.106349.

[237] A. Balanskat and K. Engelhardt, *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. 2015.

[238] J. Morales, F. Botella, C. Rusu, and D. Quiñones, 'How "Friendly" Integrated Development Environments Are?', in *Social Computing and Social Media. Design, Human Behavior and Analytics*, vol. 11578, G. Meiselwitz, Ed., in Lecture Notes in Computer Science, vol. 11578. , Cham: Springer International Publishing, 2019, pp. 80–91. doi: 10.1007/978-3-030-21902-4_7.

[239] M. Hassenzahl and N. Tractinsky, 'User experience - a research agenda', *Behav. Inf. Technol.*, vol. 25, no. 2, pp. 91–97, Mar. 2006, doi: 10.1080/01449290500330331.

[240] A. J. Ko, B. A. Myers, and H. H. Aung, 'Six Learning Barriers in End-User Programming Systems', in *2004 IEEE Symposium on Visual Languages - Human Centric Computing*, Rome, Italy: IEEE, 2004, pp. 199–206. doi: 10.1109/VLHCC.2004.47.

[241] G. C. Murphy, M. Kersten, and L. Findlater, 'How are Java software developers using the Eclipse IDE?', *IEEE Softw.*, vol. 23, no. 4, pp. 76–83, Jul. 2006, doi: 10.1109/MS.2006.105.

[242] Ibrahim, A. M, 'Correlation of Technostress Creators with Employees' Work-Life Balance in the Context of Journalists' Use of Information and Communication Technology at Work: Moderating Role of Self-Efficacy', *Int. J. Media Inf. Lit.*, vol. 6, no. 2, Dec. 2021, doi: 10.13187/ijmil.2021.2.338.

[243] D. Taser, E. Aydin, A. O. Torgaloz, and Y. Rofcanin, 'An examination of remote e-working and flow experience: The role of technostress and loneliness', *Comput. Hum. Behav.*, vol. 127, p. 107020, Feb. 2022, doi: 10.1016/j.chb.2021.107020.

[244] A. Ferziani, R. O. Rajagukguk, and P. Analya, 'Types of Technostress on Employees of IT Consulting Company', in *2018 International Conference on Orange Technologies (ICOT)*, Nusa Dua, BALI, Indonesia: IEEE, Oct. 2018, pp. 1–5. doi: 10.1109/ICOT.2018.8705838.

[245] Ben Shneiderman, Catherine Plaisant, Maxine S. Cohen, Steven M. Jacobs, and A. Marcus, 'Designing the User Interface', *Inf. Des. J.*, vol. 17, no. 2, pp. 157–158, Nov. 2009, doi: 10.1075/idj.17.2.14mar.

[246] D. A. Norman, *The design of everyday things*, Revised and Expanded ed. New York: Basic Books, 2013.

[247] D. A. Norman and S. W. Draper, *User centered system design: new perspectives on human-computer interaction*. Hillsdale (N. J.) London: L. Erlbaum, 1986.

[248] J. Preece, Y. Rogers, and H. Sharp, *Interaction design: beyond human-computer interaction*, Fourth edition. Chichester: Wiley, 2015.

[249] N. F. M. Noor, A. Saad, A. B. Ibrahim, and N. M. Noor, 'THE ACCEPTANCE OF AN EDUCATIONAL INTEGRATED DEVELOPMENT ENVIRONMENT TO LEARN PROGRAMMING FUNDAMENTALS', *Inf. Technol. Learn. Tools*, vol. 93, no. 1, pp. 135–151, Feb. 2023, doi: 10.33407/itlt.v93i1.5102.

[250] M. Aliyu, 'Exploring Learning Dynamics: A Comparative Study of IDEs and Command-Line Interfaces for Learning Java Programming language in the Department of Computer Science, Jigawa State College

of Education and Legal Studies Ringim', *Br. J. Multidiscip. Adv. Stud.*, vol. 5, no. 5, pp. 32–39, Oct. 2024, doi: 10.37745/bjmas.2022.04187.

[251] H. A. El-Sabagh, 'Adaptive e-learning environment based on learning styles and its impact on development students' engagement', *Int. J. Educ. Technol. High. Educ.*, vol. 18, no. 1, p. 53, Dec. 2021, doi: 10.1186/s41239-021-00289-4.

[252] P. W. Oman, Curtis, R. Cook, and M. Nanja, 'Effects of programming experience in debugging semantic errors', *J. Syst. Softw.*, vol. 9, no. 3, pp. 197–207, Mar. 1989, doi: 10.1016/0164-1212(89)90040-X.

[253] C. Sun, S. Yang, and B. Becker, 'Debugging in Computational Thinking: A Meta-analysis on the Effects of Interventions on Debugging Skills', *J. Educ. Comput. Res.*, vol. 62, no. 4, pp. 867–901, Jul. 2024, doi: 10.1177/07356331241227793.

[254] S. Wiedenbeck, 'Novice/expert differences in programming skills', *Int. J. Man-Mach. Stud.*, vol. 23, no. 4, pp. 383–390, Oct. 1985, doi: 10.1016/S0020-7373(85)80041-9.

[255] N. L. Schroeder, F. Yang, T. Banerjee, W. L. Romine, and S. D. Craig, 'The influence of learners' perceptions of virtual humans on learning transfer', *Comput. Educ.*, vol. 126, pp. 170–182, Nov. 2018, doi: 10.1016/j.compedu.2018.07.005.

[256] G. Rodríguez-Pérez, R. Nadri, and M. Nagappan, 'Perceived diversity in software engineering: a systematic literature review', *Empir. Softw. Eng.*, vol. 26, no. 5, p. 102, Sep. 2021, doi: 10.1007/s10664-021-09992-2.

[257] S. A. Mason and S. K. Kuttal, 'Diversity's Double-Edged Sword: Analyzing Race's Effect on Remote Pair Programming Interactions', 2024, doi: 10.48550/ARXIV.2404.07427.

[258] J. Congalton, 'A Mixed Methods Investigation of Ethnic Diversity and Productivity in Software Development Teams', 2014, *Unpublished*. doi: 10.13140/2.1.2012.3522.

[259] M. Salo, H. Pirkkalainen, C. E. H. Chua, and T. Koskelainen, 'Formation and Mitigation of Technostress in the Personal Use of IT', *MIS Q.*, vol. 46, no. 2, pp. 1073–1108, Jun. 2022, doi: 10.25300/MISQ/2022/14950.

[260] M. Valta, C. Maier, K. Pflügner, and T. Weitzel, 'Managing Technostress Across the Organisation: A Practical Framework for Business and IT Leaders', *Inf. Syst. J.*, p. isj.70005, Jul. 2025, doi: 10.1111/isj.70005.

[261] K. Pflügner, 'Technostress Management at the Workplace : A Systematic Literature Review', in *17th International Conference on Wirtschaftsinformatik*, Nürnberg: AIS Electronic Library (AISeL), 2022, p. 16. [Online]. Available: https://aisel.aisnet.org/wi2022/adoption_diffusion/adoption_diffusion/2

[262] A. Hämäläinen, M. Salo, and H. Pirkkalainen, 'Coping with technostress-related goal hindrances: multidimensional and hierarchical nature of goals', *Inf. Technol. People*, vol. 38, no. 8, pp. 222–250, Dec. 2025, doi: 10.1108/ITP-03-2024-0371.

[263] M. Berger, R. Schäfer, M. Schmidt, C. Regal, and H. Gimpel, 'How to prevent technostress at the digital workplace: a Delphi study', *J. Bus. Econ.*, vol. 94, no. 7–8, pp. 1051–1113, Oct. 2024, doi: 10.1007/s11573-023-01159-3.

[264] W. James, 'The principles of psychology. Vol. 1', New York: Cosimo Classics, 2007.

[265] R. S. Lazarus and B. N. Lazarus, *Passion and reason: making sense of our emotions*. New York Oxford: Oxford University Press, 1996.

[266] R. Plutchik, 'The Nature of Emotions', *Am. Sci.*, vol. 89, no. 4, p. 344, 2001, doi: 10.1511/2001.28.344.

[267] University of Arkansas, V. Venkatesh, S. A. Brown, University of Arizona, H. Bala, and Indiana University, 'Bridging the Qualitative-Quantitative Divide: Guidelines for Conducting Mixed Methods Research in Information Systems', *MIS Q.*, vol. 37, no. 1, pp. 21–54, Jan. 2013, doi: 10.25300/MISQ/2013/37.1.02.

[268] J. Mingers, 'Combining IS Research Methods: Towards a Pluralist Methodology', *Inf. Syst. Res.*, vol. 12, no. 3, pp. 240–259, Sep. 2001, doi: 10.1287/isre.12.3.240.9709.

[269] D. L. Morgan, 'Paradigms Lost and Pragmatism Regained: Methodological Implications of Combining Qualitative and Quantitative Methods', *J. Mix. Methods Res.*, vol. 1, no. 1, pp. 48–76, Jan. 2007, doi: 10.1177/2345678906292462.

[270] P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, Eds, *Usability Evaluation In Industry*, 0 edn. CRC Press, 1996. doi: 10.1201/9781498710411.

[271] F. D. Davis, 'Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology', *MIS Q.*, vol. 13, no. 3, p. 319, Sep. 1989, doi: 10.2307/249008.

[272] J. L. Pimentel, 'A note on the usage of Likert Scaling for research data analysis', *USM RD J.*, vol. 18, pp. 109–112, 2010.

[273] J. L. Pimentel, 'Some Biases in Likert Scaling Usage and its Correction', *Int. J. Sci. Basic Appl. Res.*, vol. 45, pp. 183–191, 2019.

[274] L. J. Cronbach, 'Coefficient Alpha and the Internal Structure of Tests', *Psychometrika*, vol. 16, no. 3, pp. 297–334, Sep. 1951, doi: 10.1007/BF02310555.

[275] A. Leontitsis and J. Pagge, 'A simulation approach on Cronbach's alpha statistical significance', *Math. Comput. Simul.*, vol. 73, no. 5, pp. 336–340, Jan. 2007, doi: 10.1016/j.matcom.2006.08.001.
[276] S. Radan, S. Hassan, A. S. Wilson, and S. S. Basurra, 'A Review of Mental Health Issues Prevalent in Science, Technology, Engineering, and Mathematics (STEM) Subjects':, in *Advances in Educational Technologies and Instructional Design*, J. Keengwe, Ed., IGI Global, 2023, pp. 1–27. doi: 10.4018/978-1-6684-9072-3.ch001.
[277] *VSCode*. (2025). Microsoft. [Online]. Available: https://code.visualstudio.com/,
[278] Stack Overflow, 'Stack Overflow Annual Developer Survey', Survey, 2025. [Online]. Available: https://survey.stackoverflow.co/
[279] J. Tan, Y. Chen, and S. Jiao, 'Visual Studio Code in Introductory Computer Science Course: An Experience Report', 2023, *arXiv*. doi: 10.48550/ARXIV.2303.10174.
[280] *VScode UI*. (2025). Microsoft. [Online]. Available: https://code.visualstudio.com/docs/getstarted/userinterface,
[281] Ian Sommerville, *SOFTWARE ENGINEERING*, 9th Edition. 2011. [Online]. Available: https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf
[282] J. Becker, N. Rush, E. Barnes, and D. Rein, 'Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity', 2025, *arXiv*. doi: 10.48550/ARXIV.2507.09089.
[283] G. Mark, D. Gudith, and U. Klocke, 'The cost of interrupted work: more speed and stress', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence Italy: ACM, Apr. 2008, pp. 107–110. doi: 10.1145/1357054.1357072.
[284] H. Sharp, Y. Rogers, and J. Preece, *Interaction design: beyond human-computer interaction*, Fifth edition. Indianapolis, IN: Wiley, 2019.
[285] womenintech, 'WOMEN IN TECH SURVEY'. 2023. [Online]. Available: https://www.womenintech.co.uk/wp-content/uploads/2022/12/Women-in-Tech-Survey-2023.pdf
[286] openai, 'Fine-tuning'. 2025. [Online]. Available: https://platform.openai.com/docs/guides/fine-tuning
[287] Microsoft, *Extensions for Visual Studio Code*. [Online]. Available: https://marketplace.visualstudio.com/
[288] *Shortcut Menu Bar*. (2018). [Universal]. Available: https://marketplace.visualstudio.com/items?itemName=jerrygoyal.shortcut-menu-bar
[289] *Pip Manager*. (2021). [Universal]. Available: https://marketplace.visualstudio.com/items?itemName=slightc.pip-manager
[290] *VSCode GitHub Copilot*. (2021). [Universal]. Available: https://marketplace.visualstudio.com/items?itemName=GitHub.copilot
[291] *Rainbow Brackets*. (2016). [Universal]. Available: https://marketplace.visualstudio.com/items?itemName=2gua.rainbow-brackets
[292] *Code Runner*. (2016). [Universal]. Available: https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner
[293] *Error Lens*. (2019). [Online]. Available: https://marketplace.visualstudio.com/items?itemName=usernamehw.errorlens
[294] Eclipse, *Eclipse Marketplace*. [Online]. Available: https://marketplace.eclipse.org/
[295] *Eclipse GitHub Copilot*. (2024). [Universal]. Available: https://marketplace.eclipse.org/content/github-copilot
[296] *Eclipse Bracketeer2*. (2024). [Universal]. Available: Universal
[297] *Eclipse Java coded templates (or code snippets)*. (2021). [Universal]. Available: https://marketplace.eclipse.org/content/java-coded-templates-or-code-snippets
[298] *Eclipse FixErrorHelper*. [Mac Only]. Available: https://marketplace.eclipse.org/content/fixerrorhelper#details
[299] JetBrains, *Plugins for JetBrains*. (2025). [Online]. Available: https://plugins.jetbrains.com/
[300] *Jetbrains Copilot*. (2025). [Online]. Available: https://plugins.jetbrains.com/plugin/17718-github-copilot
[301] *Jetbrains Rainbow Brackets*. (2025). [Universal]. Available: https://plugins.jetbrains.com/plugin/10080-rainbow-brackets
[302] Vim, *VimAwesome Plugins*. (2025). [Online]. Available: https://vimawesome.com/
[303] *Vim-snippets*. (2021). [Online]. Available: https://vimawesome.com/plugin/vim-snippets
[304] Vim, *VimAwesome Interface*. (2025). [Online]. Available: https://vimawesome.com/?q=cat%3Ainterface
[305] NPP, *nppPluginList*. (2025). [Online]. Available: https://github.com/notepad-plus-plus/nppPluginList
[306] T. S. Ragu-Nathan, M. Tarafdar, B. S. Ragu-Nathan, and Q. Tu, 'The Consequences of Technostress for End Users in Organizations: Conceptual Development and Empirical Validation', *Inf. Syst. Res.*, vol. 19, no. 4, pp. 417–433, Dec. 2008, doi: 10.1287/isre.1070.0165.

[307] R. Andrulli and R. Gerards, ‘How new ways of working during COVID-19 affect employee well-being via technostress, need for recovery, and work engagement’, *Comput. Hum. Behav.*, vol. 139, p. 107560, Feb. 2023, doi: 10.1016/j.chb.2022.107560.

[308] L. Atanasoff and M. A. Venable, ‘Technostress: Implications for Adults in the Workforce’, *Career Dev. Q.*, vol. 65, no. 4, pp. 326–338, Dec. 2017, doi: 10.1002/cdq.12111.

[309] P. Sanjeeva Kumar, ‘TECHNOSTRESS: A comprehensive literature review on dimensions, impacts, and management strategies’, *Comput. Hum. Behav. Rep.*, vol. 16, p. 100475, Dec. 2024, doi: 10.1016/j.chbr.2024.100475.

[310] iMotions, ‘What Are Emotions and Why Do They Matter?’ [Online]. Available: https://imotions.com/blog/learning/research-fundamentals/emotions-matter/?srsltid=AfmBOoqK1eXW_gT_g6n6Lrf78nP1sDt583o8xtkUUdZiAvyGhGNuqwhq

[311] I. B. Mauss and M. D. Robinson, ‘Measures of emotion: A review’, *Cogn. Emot.*, vol. 23, no. 2, pp. 209–237, Feb. 2009, doi: 10.1080/02699930802204677.

[312] J. E. LeDoux, ‘Emotion Circuits in the Brain’, *Annu. Rev. Neurosci.*, vol. 23, no. 1, pp. 155–184, Mar. 2000, doi: 10.1146/annurev.neuro.23.1.155.

[313] Y. Hellman, ‘An Innovative Partnership of Law Enforcement and Higher Education: (511532010-003)’. 2009. doi: 10.1037/e511532010-003.

[314] C. E. Izard, ‘The Many Meanings/Aspects of Emotion: Definitions, Functions, Activation, and Regulation’, *Emot. Rev.*, vol. 2, no. 4, pp. 363–370, Oct. 2010, doi: 10.1177/1754073910374661.

[315] R. B. Zajonc, ‘Feeling and thinking: Preferences need no inferences.’, *Am. Psychol.*, vol. 35, no. 2, pp. 151–175, Feb. 1980, doi: 10.1037/0003-066X.35.2.151.

[316] E. A. Phelps, ‘Emotion and Cognition: Insights from Studies of the Human Amygdala’, *Annu. Rev. Psychol.*, vol. 57, no. 1, pp. 27–53, Jan. 2006, doi: 10.1146/annurev.psych.56.091103.070234.

[317] L. Pessoa, ‘On the relationship between emotion and cognition’, *Nat. Rev. Neurosci.*, vol. 9, no. 2, pp. 148–158, Feb. 2008, doi: 10.1038/nrn2317.

[318] A. R. Damasio, *Descartes’ error: emotion, reason and the human brain*. London: Picador, 1995.

[319] P. Ekman, ‘Facial expression and emotion.’, *Am. Psychol.*, vol. 48, no. 4, pp. 384–392, 1993, doi: 10.1037/0003-066X.48.4.384.

[320] Kate Krosschell, ‘Facial Expression Analysis: The Complete Pocket Guide’. [Online]. Available: https://imotions.com/blog/learning/best-practice/facial-expression-analysis/?srsltid=AfmBOoq-O5i13ZbcRMaIsuHPnw-Ite6brJKCSFQkcKhJg7H_R5I52on6

[321] B. Nam *et al.*, ‘FacialCueNet: unmasking deception - an interpretable model for criminal interrogation using facial expressions’, *Appl. Intell.*, vol. 53, no. 22, pp. 27413–27427, Nov. 2023, doi: 10.1007/s10489-023-04968-9.

[322] J. Widacki, M. Widacki, B. Wójcik, and A. Szuba-Boroń, ‘Recognition of emotions by analysing facial expressions with FaceReader (Noldus) vs detection of deception by polygraph examination. A pilot study’, *Eur. Polygr.*, vol. 19, no. 1, pp. 31–58, Jul. 2025, doi: 10.2478/ep-2025-0002.

[323] X. Shen, G. Fan, C. Niu, and Z. Chen, ‘Catching a Liar Through Facial Expression of Fear’, *Front. Psychol.*, vol. 12, p. 675097, Jun. 2021, doi: 10.3389/fpsyg.2021.675097.

[324] M. Pantie and L. J. M. Rothkrantz, ‘Automatic analysis of facial expressions: the state of the art’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1424–1445, Dec. 2000, doi: 10.1109/34.895976.

[325] Zhihong Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, ‘A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 39–58, Jan. 2009, doi: 10.1109/TPAMI.2008.52.

[326] A. Samara, L. Galway, R. Bond, and H. Wang, ‘Affective state detection via facial expression analysis within a human–computer interaction context’, *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 6, pp. 2175–2184, Jun. 2019, doi: 10.1007/s12652-017-0636-8.

[327] S. Medjden, N. Ahmed, and M. Lataifeh, ‘Adaptive user interface design and analysis using emotion recognition through facial expressions and body posture from an RGB-D sensor’, *PLOS ONE*, vol. 15, no. 7, p. e0235908, Jul. 2020, doi: 10.1371/journal.pone.0235908.

[328] G. Tonguç and B. Ozaydın Ozkara, ‘Automatic recognition of student emotions from facial expressions during a lecture’, *Comput. Educ.*, vol. 148, p. 103797, Apr. 2020, doi: 10.1016/j.compedu.2019.103797.

[329] N. Peitek *et al.*, ‘Correlates of Programmer Efficacy and Their Link to Experience: A Combined EEG and Eye-Tracking Study’, 2023, doi: 10.48550/ARXIV.2303.07071.

[330] J. Medeiros *et al.*, ‘Can EEG Be Adopted as a Neuroscience Reference for Assessing Software Programmers’ Cognitive Load?’, *Sensors*, vol. 21, no. 7, p. 2338, Mar. 2021, doi: 10.3390/s21072338.

[331] D. Camilleri, C. Porter, and M. Micallef, ‘Assessing Task Difficulty in Software Testing using Biometric Measures’, presented at the 35th International BCS Human-Computer Interaction Conference, Jul. 2022. doi: 10.14236/ewic/HCI2022.5.

[332] J. A. Silva Da Costa and R. Gheyi, ‘Evaluating the Code Comprehension of Novices with Eye Tracking’, in *Proceedings of the XXII Brazilian Symposium on Software Quality*, Bras\'{\i}lia Brazil: ACM, Nov. 2023, pp. 332–341. doi: 10.1145/3629479.3629490.
[333] R. Bednarik and M. Tukiainen, ‘An eye-tracking methodology for characterizing program comprehension processes’, in *Proceedings of the 2006 symposium on Eye tracking research & applications - ETRA ’06*, San Diego, California: ACM Press, 2006, p. 125. doi: 10.1145/1117309.1117356.
[334] K. Kevic, B. M. Walters, T. R. Shaffer, B. Sharif, D. C. Shepherd, and T. Fritz, ‘Tracing software developers’ eyes and interactions for change tasks’, in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, Bergamo Italy: ACM, Aug. 2015, pp. 202–213. doi: 10.1145/2786805.2786864.
[335] L. Gonçales, K. Farias, B. Da Silva, and J. Fessler, ‘Measuring the Cognitive Load of Software Developers: A Systematic Mapping Study’, in *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, Montreal, QC, Canada: IEEE, May 2019, pp. 42–52. doi: 10.1109/ICPC.2019.00018.
[336] P. Ekman, ‘An argument for basic emotions’, *Cogn. Emot.*, vol. 6, no. 3–4, pp. 169–200, May 1992, doi: 10.1080/02699939208411068.
[337] S. Folkman, ‘Stress: Appraisal and Coping’, in *Encyclopedia of Behavioral Medicine*, M. D. Gellman and J. R. Turner, Eds, New York, NY: Springer New York, 2013, pp. 1913–1915. doi: 10.1007/978-1-4419-1005-9_215.
[338] S. Folkman and R. S. Lazarus, ‘An Analysis of Coping in a Middle-Aged Community Sample’, *J. Health Soc. Behav.*, vol. 21, no. 3, p. 219, Sep. 1980, doi: 10.2307/2136617.
[339] R. Pekrun, ‘The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice’, *Educ. Psychol. Rev.*, vol. 18, no. 4, pp. 315–341, Nov. 2006, doi: 10.1007/s10648-006-9029-9.
[340] M. Owens, J. Stevenson, J. A. Hadwin, and R. Norgate, ‘When does anxiety help or hinder cognitive test performance? The role of working memory capacity’, *Br. J. Psychol.*, vol. 105, no. 1, pp. 92–101, Feb. 2014, doi: 10.1111/bjop.12009.
[341] A. R. Artino and K. D. Jones, ‘Exploring the complex relations between achievement emotions and self-regulated learning behaviors in online learning’, *Internet High. Educ.*, vol. 15, no. 3, pp. 170–175, Jun. 2012, doi: 10.1016/j.iheduc.2012.01.006.
[342] G. Noteborn, K. Bohle Carbonell, A. Dailey-Hebert, and W. Gijselaers, ‘The role of emotions and task significance in Virtual Education’, *Internet High. Educ.*, vol. 15, no. 3, pp. 176–183, Jun. 2012, doi: 10.1016/j.iheduc.2012.03.002.
[343] R. Pekrun, A. C. Frenzel, T. Goetz, and R. P. Perry, ‘The Control-Value Theory of Achievement Emotions’, in *Emotion in Education*, Elsevier, 2007, pp. 13–36. doi: 10.1016/B978-012372545-5/50003-4.
[344] R. Pekrun, T. Goetz, A. C. Frenzel, P. Barchfeld, and R. P. Perry, ‘Measuring emotions in students’ learning and performance: The Achievement Emotions Questionnaire (AEQ)’, *Contemp. Educ. Psychol.*, vol. 36, no. 1, pp. 36–48, Jan. 2011, doi: 10.1016/j.cedpsych.2010.10.002.
[345] R. Pekrun and E. J. Stephens, ‘Achievement Emotions: A Control-Value Approach: Achievement Emotions’, *Soc. Personal. Psychol. Compass*, vol. 4, no. 4, pp. 238–255, Apr. 2010, doi: 10.1111/j.1751-9004.2010.00259.x.
[346] J. A. Russell, ‘A circumplex model of affect.’, *J. Pers. Soc. Psychol.*, vol. 39, no. 6, pp. 1161–1178, Dec. 1980, doi: 10.1037/h0077714.
[347] J. Posner, J. A. Russell, and B. S. Peterson, ‘The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology’, *Dev. Psychopathol.*, vol. 17, no. 03, Sep. 2005, doi: 10.1017/S0954579405050340.
[348] L. F. Barrett, *Seven and a half lessons about the brain*, First Mariner Books edition. Boston New York: Mariner Books, an imprint of HarperCollins Publishers, 2021.
[349] L. F. Barrett, ‘The theory of constructed emotion: an active inference account of interoception and categorization’, *Soc. Cogn. Affect. Neurosci.*, p. nsw154, Oct. 2016, doi: 10.1093/scan/nsw154.
[350] J. H. Turner, *Human emotions: a sociological theory*, 1. publ. London: Routledge, 2007.
[351] B. L. Fredrickson, ‘The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions.’, *Am. Psychol.*, vol. 56, no. 3, pp. 218–226, Mar. 2001, doi: 10.1037/0003-066X.56.3.218.
[352] A. B. Burns *et al.*, ‘Upward spirals of positive emotion and coping: Replication, extension, and initial exploration of neurochemical substrates’, *Personal. Individ. Differ.*, vol. 44, no. 2, pp. 360–370, Jan. 2008, doi: 10.1016/j.paid.2007.08.015.

[353] B. L. Fredrickson and T. Joiner, ‘Positive Emotions Trigger Upward Spirals Toward Emotional Well-Being’, *Psychol. Sci.*, vol. 13, no. 2, pp. 172–175, Mar. 2002, doi: 10.1111/1467-9280.00431.
[354] A. S. P. Jansen, X. Van Nguyen, V. Karpitskiy, T. C. Mettenleiter, and A. D. Loewy, ‘Central Command Neurons of the Sympathetic Nervous System: Basis of the Fight-or-Flight Response’, *Science*, vol. 270, no. 5236, pp. 644–646, Oct. 1995, doi: 10.1126/science.270.5236.644.
[355] *iMotions Lab*. (2025). [Online]. Available: https://imotions.com/products/imotions-lab/
[356] iMotions, *iMotions EduLabs*. (2025). [Online]. Available: https://imotions.com/edulabs/
[357] ‘iMotions Publications’. 2025. [Online]. Available: https://imotions.com/blog/publications/
[358] Y.-Q. Wang, ‘An Analysis of the Viola-Jones Face Detection Algorithm’, *Image Process. Line*, vol. 4, pp. 128–148, Jun. 2014, doi: 10.5201/ipol.2014.104.
[359] M. Bishay, K. Preston, M. Strafuss, G. Page, J. Turcot, and M. Mavadati, ‘AFFDEX 2.0: A Real-Time Facial Expression Analysis Toolkit’, 2022, *arXiv*. doi: 10.48550/ARXIV.2202.12059.
[360] J. J. Gibson, *The ecological approach to visual perception*. Hillsdale (N.J.) London: L. Erlbaum, 1986.
[361] S. K. D’Mello, B. Lehman, and N. Person, ‘Monitoring Affect States During Effortful Problem Solving Activities’, *Int. J. Artif. Intell. Educ.*, vol. 20, no. 4, pp. 361–389, Nov. 2010, doi: 10.3233/JAI-2010-012.
[362] N. Bosch and S. D’Mello, ‘The Affective Experience of Novice Computer Programmers’, *Int. J. Artif. Intell. Educ.*, vol. 27, no. 1, pp. 181–206, Mar. 2017, doi: 10.1007/s40593-015-0069-5.
[363] A. Arguel, L. Lockyer, G. Kennedy, J. M. Lodge, and M. Pachman, ‘Seeking optimal confusion: a review on epistemic emotion management in interactive digital learning environments’, *Interact. Learn. Environ.*, vol. 27, no. 2, pp. 200–210, Feb. 2019, doi: 10.1080/10494820.2018.1457544.
[364] M. Wrobel, ‘Applicability of Emotion Recognition and Induction Methods to Study the Behavior of Programmers’, *Appl. Sci.*, vol. 8, no. 3, p. 323, Feb. 2018, doi: 10.3390/app8030323.
[365] Y. Wang *et al.*, ‘A Systematic Review on Affective Computing: Emotion Models, Databases, and Recent Advances’, 2022, *arXiv*. doi: 10.48550/ARXIV.2203.06935.
[366] J. McCambridge, J. Witton, and D. R. Elbourne, ‘Systematic review of the Hawthorne effect: New concepts are needed to study research participation effects’, *J. Clin. Epidemiol.*, vol. 67, no. 3, pp. 267–277, Mar. 2014, doi: 10.1016/j.jclinepi.2013.08.015.
[367] J. Sarabadani, D. Compeau, and M. Carter, ‘An Investigation of IT Users’ Emotional Responses to Technostress Creators’, presented at the Hawaii International Conference on System Sciences, 2020. doi: 10.24251/HICSS.2020.748.
[368] Beaudry and Pinsonneault, ‘The Other Side of Acceptance: Studying the Direct and Indirect Effects of Emotions on Information Technology Use’, *MIS Q.*, vol. 34, no. 4, p. 689, 2010, doi: 10.2307/25750701.
[369] I. A. Khan, W.-P. Brinkman, and R. M. Hierons, ‘Do moods affect programmers’ debug performance?’, *Cogn. Technol. Work*, vol. 13, no. 4, pp. 245–258, Nov. 2011, doi: 10.1007/s10111-010-0164-1.
[370] S. C. Muller 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*, Florence, Italy: IEEE, May 2015, pp. 688–699. doi: 10.1109/ICSE.2015.334.
[371] D. Graziotin, X. Wang, and P. Abrahamsson, ‘Happy software developers solve problems better: psychological measurements in empirical software engineering’, *PeerJ*, vol. 2, p. e289, Mar. 2014, doi: 10.7717/peerj.289.
[372] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, ‘What happens when software developers are (un)happy’, *J. Syst. Softw.*, vol. 140, pp. 32–47, Jun. 2018, doi: 10.1016/j.jss.2018.02.041.
[373] D. Girardi, F. Lanubile, N. Novielli, and A. Serebrenik, ‘Emotions and Perceived Productivity of Software Developers at the Workplace’, *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3326–3341, Sep. 2022, doi: 10.1109/TSE.2021.3087906.
[374] X. Ge, ‘Emotion matters for academic success: Implications of the Article by Jarrell, Harley, Lajoie, and Naismith (2017) for creating nurturing and supportive learning environments to help students manage their emotions’, *Educ. Technol. Res. Dev.*, vol. 69, no. 1, pp. 67–70, Feb. 2021, doi: 10.1007/s11423-020-09925-8.
[375] A. P. Brief and H. M. Weiss, ‘Organizational Behavior: Affect in the Workplace’, *Annu. Rev. Psychol.*, vol. 53, no. 1, pp. 279–307, Feb. 2002, doi: 10.1146/annurev.psych.53.100901.135156.
[376] W. Burleson and Rosalind W Picard, ‘Affective agents: Sustaining motivation to learn through failure and a state of stuck’, *Workshop Soc. Emot. Intell. Learn. Environ.*, no. 7th conference on intelligent tutoring systems, Maceió, Brasil., 2004, [Online]. Available: https://scholar.google.com/scholar_lookup?title=Affective+agents%3A+Sustaining+motivation+to+learn+through+failure+and+a+state+of+%E2%80%9Cstuck.%E2%80%9D&author=W.+Burleson&author=R.+W.+Picard&publication_year=2004
[377] M. Tavakol and R. Dennick, ‘Making sense of Cronbach’s alpha’, *Int. J. Med. Educ.*, vol. 2, pp. 53–55, Jun. 2011, doi: 10.5116/ijme.4dfb.8dfd.

[378] R. A. L. F. Van Griethuijsen *et al.*, ‘Global Patterns in Students’ Views of Science and Interest in Science’, *Res. Sci. Educ.*, vol. 45, no. 4, pp. 581–603, Aug. 2015, doi: 10.1007/s11165-014-9438-6.

[379] R. F. DeVellis, *Scale development: theory and applications*, Fourth edition. in Applied social research methods series, no. 26. Los Angeles, Calif. London New Delhi Singapore Washington, DC Melbourne: SAGE, 2017.

[380] A. Landowska and J. Miler, ‘Limitations of Emotion Recognition in Software User Experience Evaluation Context’, presented at the 2016 Federated Conference on Computer Science and Information Systems, Oct. 2016, pp. 1631–1640. doi: 10.15439/2016F535.

[381] B. Paudyal, C. Creed, I. Williams, and M. Frutos-Pascual, ‘Inclusive Multimodal Voice Interaction for Code Navigation’, in *Proceedings of the 2022 International Conference on Multimodal Interaction*, Bengaluru India: ACM, Nov. 2022, pp. 509–519. doi: 10.1145/3536221.3556600.

[382] L. Halpin, P. Benachour, T. Hall, A.-M. Houghton, and E. Winter, ‘Accessible Design in Integrated Development Environments: A Think Aloud Study Exploring the Experiences of Students with ADHD’, 2025, *arXiv*. doi: 10.48550/ARXIV.2506.10598.

[383] K. Dilmen, S. B. Kert, and T. Uğraş, ‘Children’s coding experiences in a block-based coding environment: a usability study on code.org’, *Educ. Inf. Technol.*, vol. 28, no. 9, pp. 10839–10864, Sep. 2023, doi: 10.1007/s10639-023-11625-8.

8 Appendix

The appendices included in this thesis provide supplementary material that supports and extends the main body of the research.

8.1 Part A – Consent Form

Section 1 – Overview

Title: Well-being Integrated Development Environments (WIDEs)

Summary: This project aims to understand if IDEs which are used for programming can be designed such that they induce less mental stress on users (i.e., students).

Section 2 – Statement about Voluntary Participation

Participation in this project as outlined in the participant information sheet is entirely voluntary. Non-participation is entirely your choice and will not have any adverse effects on your study, degree outcome or your time at BCU. If you require alternative means of explaining the project due you written communication issues, please contact sasan.radan@bcu.ac.uk to arrange that.

Section 3 - Consent

☐ Read and understood the Participant Information Sheet.

☐ Have had the opportunity to ask questions.

☐ Understand that participation is entirely voluntary.

☐ Agree to take part in the controlled experiment outlined in the Participant Information Sheet.

☐ Agree to answer survey questions about your experience in using the IDE as part of the controlled experiment

☐ Agree to be observed using facial expression, eye-motion, and mouse-clicking tracking sensors/software while using the IDE in the controlled experiment

☐ Understand that you have the right withdraw (details in the Participant Information Sheet) without prejudice.

☐ Understand your right to anonymity/confidentiality (for example where anonymous quotes will be used in publications).

☐ Understand that if you decide to withdraw as outlined in the Participant Information Sheet, your contribution to the project will be destroyed and it will not be used to conclude any results in the project.

☐ Understand that withdrawal during the controlled experiment is allowed.

Section 4 – Demographic

[Q1] Gender? (Male, Female, Other, Prefer Not to Say, Other)
[Q2] Age Group? (Under 18 Years, 18 – 25 Years, Over 25 Years)
[Q3] Year of Study? (First Year, Second Year, Third Year, Masters, PhD)
[Q4] Status? (Home/EU student, International Student, Prefer Not to Say)
[Q5] Ethnicity? (Asian, Black, White, Other (Please Specify))
[Q6] Disabilities? (Yes (Please Specify), No, Prefer Not to Say, Other (Please Specify))
[Q7] Programming Experience? (Under 1 Year, 1 – 2 Years, Over 2 Years)

Section 4 – Declaration

[Q1] Enter Full Name (insert name)
[Q2] Enter Institution Email (insert email)
[Q3] Signature (enter name)
[Q4] Date of Signature (select date)

Participants consent form (used as part of Chapter 3) to partake in the experiment.

Note: Age (Q2), nationality (Q4) and disability (Q6) are included in this consent form to provide a comprehensive overview of participant characteristics. These variables were not incorporated into the statistical analyses presented in Chapter 3 due to limited subgroup representation and because they were not part of the predefined demographic comparison criteria

8.2 Part B – Coding Tasks

8.2.1 Task A

Write a program to check university students' degree classification. Your program should fulfil the following conditions.

- Average mark less than 40% is a Fail.
- Average mark greater than or equal to 40% is a Pass.

For passing students, using the table below also identify the student's degree classification based on the average mark achieved.

Classification	Mark
First-class (1st)	70%+
Upper second-class (2.1)	60-69%
Lower second-class (2.2)	50-59%
Third-class (3rd)	40-49%

When developing your solution, incorporate:

- Dictionary[s]
- Error handling
- If…else statement[s]
- Function[s]
- Operator[s]
- Format string[s]
- User Input

Also, make use of the following given dictionary:

- classif_dict = {"first" : "First-class (1st)", "second": "Second-class (2nd)", "third" : "Third-class (3rd)"}

Examples of the expected input/outputs provided below

```
Enter name: sasan
Enter Module A mark: 72
Enter Module B mark: 68
Enter Module C mark: 82

--------- Undergrad Degree Classification -------------
Name: Sasan
Average: 74.00%
Status: Passed Degree!
Classification: First-class (1st)
```

```
Enter name: jake
Enter Module A mark: 25
Enter Module B mark: 37
Enter Module C mark: 36

--------- Undergrad Degree Classification -------------
Name: Jake
Average: 32.67%
Status: Failed Degree!
```

```
Enter name: sara
Enter Module A mark: 72
Enter Module B mark: 40
Enter Module C mark: 42

--------- Undergrad Degree Classification -------------
Name: Sara
Average: 51.33%
Status: Passed Degree!
Classification: Lower second-class (2.2)
```

```
Enter name: harry
Enter Module A mark: -10
Enter Module B mark: -50
Enter Module C mark: -30

--------- Undergrad Degree Classification -------------
Name: Harry
Average: -30.00%
Error: percentage should be between 0 and 100 only!
```

8.2.2 Task B

Write a program to create a holiday expense calculator which works out the: hotel, plane, car rental and total trip cost based on:

(a) Number of nights staying at hotel
(b) Chosen destination city
(c) Car rental period (if 3 or more days £20 off, if 7 or more days $50 off)
(d) Spending money

When developing your solution, incorporate:

- Dictionary[s]
- Error handling
- If…else statement[s]
- Function[s]
- Operator[s]
- Format string[s]
- User Input

Also, make use of:

- hotel_rate = 150
- rental_rate = 40
- cities = {"Rome":200, "Madrid":300, "Athens":500,"Tokyo":800}

Examples of the expected input/outputs provided below

```
Enter nights: 3
Enter City (Rome, Madrid, Athens, Tokyo): Rome
Enter days of car rental: 3
Spending money: 200


___________________ Holiday Expense Calculator ___________________

Hotel Cost      | 3 nights x 150 (rate) = £450

Plane Cost      | Rome (one way) = £200

Car Hire Info   | 3 or more days £20 off
                | 7 or more days £50 off
Rental Car Cost | 3 day hire x £40 (rate) = £100

Spending Money  | £200.00

Trip Cost       | £950.00
```

```
Enter nights: 9
Enter City (Rome, Madrid, Athens, Tokyo): Madrid
Enter days of car rental: 8
Spending money: 200


___________________ Holiday Expense Calculator ___________________

Hotel Cost      | 9 nights x 150 (rate) = £1350

Plane Cost      | Madrid (one way) = £300

Car Hire Info   | 3 or more days £20 off
                | 7 or more days £50 off
Rental Car Cost | 8 day hire x £40 (rate) = £270

Spending Money  | £200.00

Trip Cost       | £2120.00
```

```
Enter nights: 4
Enter City (Rome, Madrid, Athens, Tokyo): Tokyo
Enter days of car rental: 4
Spending money: 500


___________________ Holiday Expense Calculator ___________________

Hotel Cost      | 4 nights x 150 (rate) = £600

Plane Cost      | Tokyo (one way) = £800

Car Hire Info   | 3 or more days £20 off
                | 7 or more days £50 off
Rental Car Cost | 4 day hire x £40 (rate) = £140

Spending Money  | £500.00

Trip Cost       | £2040.00
```

```
Enter nights: 6
Enter City (Rome, Madrid, Athens, Tokyo): Athens
Enter days of car rental: 3
Spending money: 100


___________________ Holiday Expense Calculator ___________________

Hotel Cost      | 6 nights x 150 (rate) = £900

Plane Cost      | Athens (one way) = £500

Car Hire Info   | 3 or more days £20 off
                | 7 or more days £50 off
Rental Car Cost | 3 day hire x £40 (rate) = £100

Spending Money  | £100.00

Trip Cost       | £1600.00
```

```
Enter nights: xxx
Invalid Input

Enter nights: 2
Enter City (Rome, Madrid, Athens, Tokyo): xxx
Invalid City Selected

Enter nights: 2
Enter City (Rome, Madrid, Athens, Tokyo): Rome
Enter days of car rental: 2
Spending money: 100


___________________ Holiday Expense Calculator ___________________

Hotel Cost      | 2 nights x 150 (rate) = £300

Plane Cost      | Rome (one way) = £200

Car Hire Info   | 3 or more days £20 off
                | 7 or more days £50 off
Rental Car Cost | 2 day hire x £40 (rate) = £80

Spending Money  | £100.00

Trip Cost       | £680.00
```