Journal of Information Systems Education

Volume 37
Issue 4
Fall 2026

Supporting Introductory Python Learning Through Graphical Output: Evaluating Thonny-py5mode Using Task-Technology Fit

Tristan Bunn, Aaron Bere, and Michelle Douglas

Recommended Citation: Bunn, T., Bere, A., & Douglas, M. (2026). Supporting Introductory Python Learning Through Graphical Output: Evaluating Thonny-py5mode Using Task-Technology Fit. *Journal of Information Systems Education*, 37(4), 546-565. https://doi.org/10.62273/RDCV4243

Article Link: https://jise.org/Volume37/n4/JISE2026v37n4pp546-565.html

Received:	October 29, 2025
First Decision:	November 29, 2025
Accepted:	February 18, 2026
Published:	December 15, 2026

Find archived papers, submission instructions, terms of use, and much more at the JISE website:
https://jise.org

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 37(4), 546-565, Fall 2026
https://doi.org/10.62273/RDCV4243

Supporting Introductory Python Learning Through Graphical Output: Evaluating Thonny-py5mode Using Task-Technology Fit

Tristan Bunn
Department of Design & Creative Technology
Torrens University Australia
Surry Hills Campus, Sydney, Australia
tristan.bunn@torrens.edu.au

Aaron Bere
Department of Design & Creative Technology
Torrens University Australia
Flinders Street Campus, Melbourne, Australia
aaron.bere@torrens.edu.au

Michelle Douglas
School of Design
RMIT University Australia
City Campus, Melbourne, Australia
michelle.douglas@rmit.edu.au

ABSTRACT

Coding environments that integrate graphical output into programming instruction can enhance engagement, comprehension, and motivation among beginners. This study evaluates Thonny-py5mode: a plugin for the Thonny IDE that emulates the Java-based Processing creative coding IDE but instead supports Python via the py5 library. Guided by the Task-Technology Fit (TTF) framework, we extend the model with personalised learning, hedonic motivation, and effort expectancy to capture factors influencing Thonny-py5mode user adoption in educational contexts. The study analyses survey data from 143 students in a first-year university Python course that incorporated four weeks of Thonny-py5mode use, including an assessment requiring participants to recreate six graphical challenges. We employ Structural Equation Modelling (SEM) to test nine hypotheses mapped to our extended TTF framework. Our results show that Thonny-py5mode's usability and task support, along with its ease of use, contributed to a strong perceived TTF, which in turn influenced students' intention to continue using it. These results highlight the Thonny-py5mode design priorities that can encourage adoption and sustained use in related educational coding environments and tools.

Keywords: Task technology fit (TTF), Python, Introductory programming, Integrated development environment (IDE)

1. INTRODUCTION

Introductory programming courses commonly prioritise text-based input and console output, emphasising syntax and abstract logic over visual or interactive feedback. While this approach aligns with traditional computer science pedagogy, it can pose barriers for novice learners, particularly those from creative or interdisciplinary backgrounds who may struggle to connect abstract code to observable outcomes. Prior research indicates that programming for graphical output can reduce cognitive load, encourage experimentation, and increase learner motivation by making code behaviour more visible and tangible (Chiodini et al., 2025).

Creative-coding environments such as Processing and p5.js address these challenges by enabling learners to generate graphical and interactive outputs with minimal setup code (Funk & Zhang, 2024; Shiffman, 2024). In contrast to console-oriented environments, these conveniently packaged beginner-friendly IDEs (similar to Figure 1) centre on graphical output rather than textual feedback. In Processing, a persistent drawing canvas controls a continuous animation loop (via a *draw()* function), with easy programmatic access to built-in mouse and keyboard event handling. To simplify things further, beginners can omit the *draw()* function entirely and enter simple *circle()*, *square()*, and other graphics functions without animating them—and, in this way, decipher the workings of their code expressed through visible shapes, motion, and colour, as opposed to inferring programming state and dynamics from textual output alone. That said, Processing and p5.js also include a console that supports debugging and print-based output alongside visuals.

However, these environments rely on Java or JavaScript, creating misalignment for courses that prioritise Python due to its readability, low entry threshold, and cross-disciplinary relevance (Guo, 2021; Xu & Frydenberg, 2021). Processing 3.x previously offered a Python mode, Processing.py, but it is now unmaintained and incompatible with Processing 4.x, leaving a gap for Python–Processing pedagogy (Feinburg, 2023). This study addresses that gap through Thonny-py5mode: a bespoke plugin that integrates the Processing creative-coding paradigm using the beginner-oriented Thonny IDE and the py5 graphics library.

Thonny-py5mode (Figure 1) allows students to write Processing-style sketches in Python without manual package installation or environment configuration. The interface mirrors the Processing IDE, including a dedicated sketch runner and additional Processing-inspired tools (like a colour mixer) to reduce friction for novice learners or those transitioning to Python from other Processing environments.

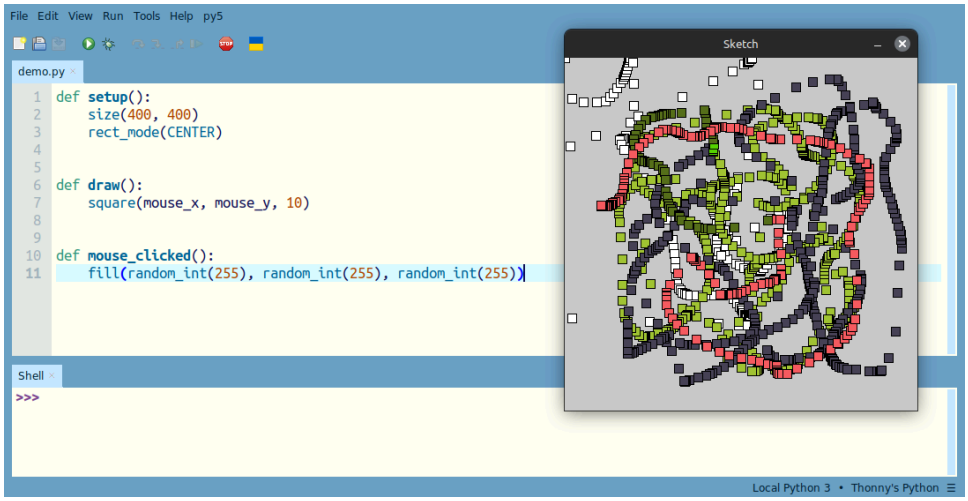


Figure 1. Thonny-py5mode Interface

While tool development for novice programmers is well established, fewer studies empirically examine how Processing-like environments align with graphical programming tasks and learners' personal learning characteristics. To respond to this need, this study evaluates Thonny-py5mode using the Task–Technology Fit (TTF) framework (Goodhue & Thompson, 1995). TTF posits that technology use and performance depend on how well a system supports the tasks users must perform. Recognising critiques that traditional TTF applications underrepresent learner-level factors (Tzeng et al., 2022), we extend the framework by incorporating effort expectancy, hedonic motivation, and personalised learning as characteristics that influence perceived fit and adoption. Guided by this extended TTF framework, we report findings from a survey of students enrolled in an introductory Python programming course, investigating how well Thonny-py5mode supported their programming tasks by reducing perceived effort, accommodating learner preferences, and enhancing engagement through graphical output.

The rest of this paper is structured as follows: Section 2 reviews prior work on introductory programming environments for graphical output and the theoretical models relevant to the study. Section 3 (Hypothesis Development) describes our extended TTF framework, including formal definitions for personalised learning, hedonic motivation, and effort expectancy. Section 4 describes the course context and instructional setting; Section 5 details research methodology, instrument design, data collection, and analytical approach. Section 6 presents the SEM analysis results and hypothesis testing, with Section 7 discussing how the findings relate to theory and prior research. Section 8 outlines the study's implications for research and practice, while Section 9 (Conclusion) summarises key contributions, limitations, and directions for future work.

This paper makes two key contributions: (1) it reports an empirical examination of Thonny-py5mode as a Python-based creative-coding environment used by novice learners in an introductory programming course; and (2) it applies and extends the TTF framework by incorporating learner personal characteristics to analyse students' perceptions and intention to continue using the tool.

2. BACKGROUND

Learning to program requires novices to develop syntactic knowledge, abstract reasoning, and mental models of program execution. In text-only programming contexts, learners must infer program behaviour from code and console output, which can impose higher cognitive demands and hinder early understanding (Chiodini et al., 2025). Prior studies show that programming graphical output supports experimentation, reduces cognitive load, and increases engagement, particularly for beginners and those from creative or interdisciplinary backgrounds (Bakar et al., 2020; Chiodini et al., 2025).

Python is widely adopted in introductory programming due to its readable syntax, comparatively shallow learning curve, and relevance across various domains (Guo, 2021; Xu & Frydenberg, 2021). Despite this, many Python learning settings rely primarily on console-based interaction, limiting opportunities for visual feedback. Creative-coding environments such as Processing and p5.js demonstrate how simplified graphical frameworks can support novice programming tasks (Funk & Zhang, 2024; Shiffman, 2024), but their reliance on Java or JavaScript restricts their integration into Python-focused curricula. While Python's Turtle module offers basic graphical capabilities, it does not readily support more expressive or interactive creative-coding tasks (Hunt, 2019); Pygame provides robust support for interactive graphics but primarily orients toward 2D game development rather than rapid visual sketching and Processing-style pedagogy (Hussain & Hussain, 2024).

The TTF model proposes that we can maximise technology use and performance when system capabilities align with task requirements (Goodhue & Thompson, 1995). Although previous research has proven that the fit between learning task requirements and technological characteristics is a central antecedent of technology adoption, these studies overlooked students' personal characteristics. This omission weakened the TTF's effectiveness and reduced its explanatory power, as technology must not only meet task requirements but also align with individual learners' traits (Bere, 2018). Researchers have observed similar limitations in TPB (Theory of Planned Behaviour), which emphasises attitude, subjective norm, and perceived behavioural control, but has faced criticism for neglecting individual-level

characteristics beyond its core constructs (Tzeng et al., 2022). Beyond generic adoption models, studies in South African higher and basic education underscore that curriculum-level technology integration and educators' ICT self-efficacy meaningfully shape how they appropriate tools in practice (Mlambo et al., 2020; Rambe, 2016).

3. HYPOTHESIS DEVELOPMENT

Figure 2 presents our extended TTF model, integrating the nine hypotheses. To better frame these, we broaden the *Personal Characteristics* construct.

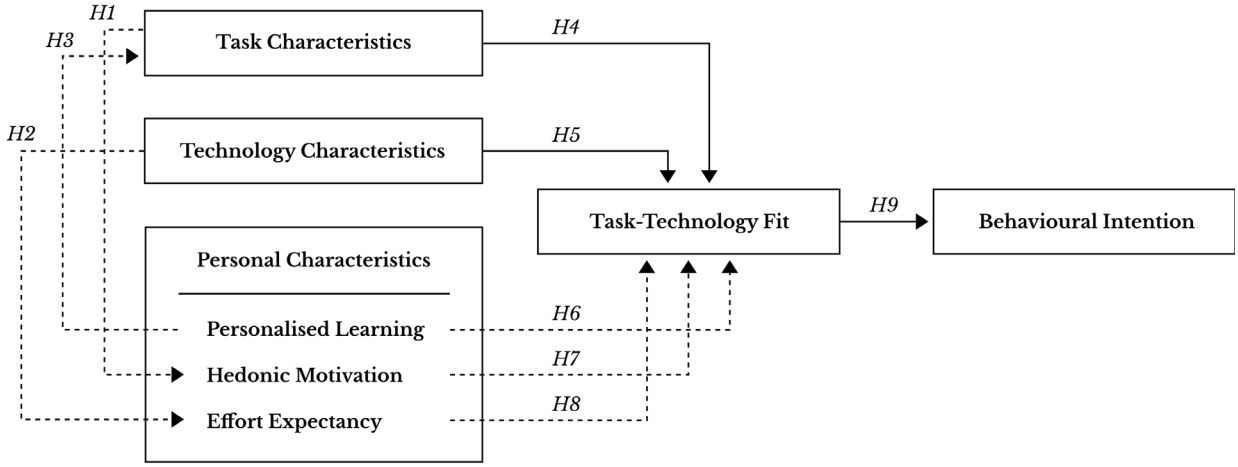


Figure 2. Extended TTF Model for Thonny-py5mode

Specifically, the Personal Characteristics domain includes *Personalised Learning*, *Hedonic Motivation*, and *Effort Expectancy*—factors shown to influence learners' technology engagement and capture individual differences in perceived fit (Shahzad et al., 2025). This highlights the pedagogical affordances of Thonny-py5mode that go beyond functional alignment, incorporating insights from acceptance models such as TAM and UTAUT/2, in which tailored experiences, perceived enjoyment, and ease of use are recognised as critical drivers of behavioural intention (Bere & Rambe, 2016; Mosunmola et al., 2018).

In programming education, tasks are not experienced solely as functional requirements but are interpreted through learners' cognitive capacity, motivation, and perceived effort. Accordingly, these personal characteristics (personalised learning, hedonic motivation, effort expectancy) are theorised to explain how task characteristics and technology characteristics translate into perceived task–technology fit. This extension aligns with prior critiques (Bere, 2018; Lim & Lee, 2021) that TTF underrepresents learner-level factors and responds to calls for models that better capture affective and experiential dimensions of technology use in learning environments. This conceptualisation is grounded in the course design, which required students to recreate predefined graphical challenges using a constrained set of Thonny-py5 drawing functions, Python control structures and loops. We detail the conceptualisation and operationalisation of all the constructs in the subsequent sections, where we develop their specific hypotheses.

RQ1: How well does Thonny-py5mode support introductory Python programming tasks in terms of task–technology fit?

RQ2: How do learner personal characteristics (personalised learning, hedonic motivation, and effort expectancy) influence perceived fit and intention to use the tool?

Sections 3.1–3.8 following describe each construct and its associated hypotheses (Figure 2), while Section 4 details the analysis methodology.

3.1 Task Characteristics

In higher education, task characteristics refer to the cognitive and instructional demands of learning activities such as problem complexity, structure, collaboration, and cognitive load that influence how effectively technology supports student learning (Bere, 2018). In the TTF model, better alignment between task requirements and technology capabilities leads to improved performance, satisfaction, and adoption (Goodhue & Thompson, 1995; Hidayat et al., 2021). In this study, Python programming tasks include writing and debugging code, incorporating py5 drawing functions, and solving graphical challenges, which require rapid feedback, low cognitive load, and ease of use—capabilities Thonny and Thonny-py5mode strive to provide. We base the following hypotheses on the task characteristics relationship with hedonic motivation and TTF constructs.

H1: Task Characteristics positively influence the Hedonic Motivation associated with Thonny-py5mode-supported learning.

H4: Task Characteristics positively influence the Task–Technology Fit associated with Thonny-py5mode-supported learning.

3.2 Technology Characteristics

Refers to the functional features, properties, and usability aspects of a system that determine how effectively it supports users' tasks (Goodhue & Thompson, 1995), including reliability, accessibility, data quality, interactivity, and alignment with user abilities and needs (DeLone & McLean, 2003). In the TTF framework, strong alignment between technology characteristics and task requirements enhances performance, adoption, and satisfaction, while poor alignment can hinder task completion, particularly in education (Park, 2019). In this study, Thonny-py5mode's beginner-friendly interface, graphical output, and real-time debugging aim to meet the needs of Python novices, lowering cognitive load and supporting tasks such as py5 coding, visualisation, and problem-solving. We base the following hypotheses on the technology characteristics construct relationship with effort expectancy and task–technology fit, respectively.

H2: Technology Characteristics positively influence the Effort Expectancy associated with Thonny-py5mode-supported learning.

H5: Technology Characteristics positively influence the Task–Technology Fit associated with Thonny-py5mode-supported learning.

3.3 Personal Characteristics

Refers to individual user traits, motivations, preferences, and cognitive capabilities that can affect how a user perceives and interacts with technology, as well as how they perform tasks (Furneaux, 2012; Venkatesh et al., 2003). These attributes shape perceptions of fit, ease of use, and enjoyment, and can ultimately affect technology adoption and outcomes (Tamilmani et al., 2021; Wang & Kartika Sari, 2024). As Figure 2 illustrates, we operationalise personal characteristics through three constructs: *Personalised Learning*, *Hedonic Motivation*, and *Effort Expectancy*, detailed in the subsections 3.4, 3.5, and 3.6, respectively.

3.4 Personalised Learning

Refers to tailoring the environment to an individual's needs, preferences, and pace of learning to enhance engagement, satisfaction, and performance (Melzer, 2019). In TTF terms, personalisation enables system features and tasks to adapt to learners' cognitive abilities, motivations, and learning styles. Thonny-py5mode supports this through optional tool panels, real-time visual feedback, and coding activities that can be approached at varying levels of complexity, allowing learners to progress at their own speed and focus on relevant concepts. By aligning with students' cognitive and emotional needs, these features can increase engagement and enjoyment in programming, particularly for novices (Alyoussef, 2021; Škorić et al., 2021). In resource-constrained contexts, higher ICT self-efficacy supports more constructive classroom uses of technology, suggesting that personalisation, which builds confidence, can strengthen fit (Mlambo et al., 2020). We base the following hypotheses on the personalised learning construct relationship with task characteristics and task–technology fit, respectively.

H3: Personalised Learning positively influences the Task Characteristics associated with Thonny-py5mode-supported learning.

H6: Personalised Learning positively influences the Task–Technology Fit associated with Thonny-py5mode-supported learning.

3.5 Hedonic Motivation

Refers to the extent to which a technology is perceived as enjoyable or fun to use (Ayasrah, 2020). Higher enjoyment can increase users' intention to adopt the technology and lead to their improved task performance (Ayasrah, 2020; Cheng et al., 2020). Within the TTF framework, hedonic motivation can also promote engagement and reduce cognitive resistance in learning contexts (Thabet et al., 2024). In this study, Thonny-py5mode's visual feedback, interactive coding environment, and immediacy aim to create an enjoyable learning experience that sustains interest—particularly for novice programmers—enhancing satisfaction and encouraging continued use (Wang & Kartika Sari, 2024). We base the following hypothesis on the relationship between the hedonic motivation construct and task–technology fit.

H7: Hedonic Motivation positively influences the Task–Technology Fit associated with Thonny-py5mode-supported learning.

3.6 Effort Expectancy

Refers to the perceived ease of use of a technology (Ayasrah, 2020). Alongside TTF, it is a key construct for user acceptance and can influence performance in educational contexts (Alyoussef, 2021; Škorić et al., 2021). TTF suggests that the easier a technology is to use, the better its fit to the task, the stronger the learning outcomes, and the lower the resistance to adoption (Abbad, 2021). In this study, Thonny-py5mode is designed to offer high perceived ease-of-use through a simplified, learner-friendly interface, clear error messages, and near-instant visual feedback. These features minimise the effort required for novices to complete coding tasks effectively and confidently (Annamaa, 2024; Škorić et al., 2021). We base the following hypothesis on the effort expectancy construct relationship with task–technology fit.

H8: Effort Expectancy positively influences the Task–Technology Fit associated with Thonny-py5mode-supported learning.

3.7 Task–Technology Fit

Describes the degree of alignment between a technology's capabilities and the requirements of a user's task (Goodhue & Thompson, 1995; Wang & Kartika Sari, 2024). As Figure 2 illustrates, it encompasses three dimensions—*Task Characteristics*, *Technology Characteristics*, and *Personal Characteristics*—that together determine how effectively a system supports task performance. High fit is typically associated with positive outcomes such as productivity, satisfaction, and behavioural intention to use the technology. In this study, Thonny-py5mode-supported learning demonstrates task–technology fit when these three dimensions align, meeting the demands of graphical Python programming tasks and the needs of novice learners. We base the following hypothesis on the task–technology fit construct relationship with behavioural intention.

H9: Task–technology fit positively influences students' Behavioural Intention to learn using Thonny-py5mode.

3.8 Behavioural Intention

Refers to a person's willingness or intent to use a technology in the future, over a sustained period (Thabet et al., 2024; Venkatesh et al., 2003). This is a core psychological variable in technology acceptance models such as UTAUT and UTAUT2, where it not only reflects interest but also predicts actual system use. Several aspects shape behavioural intention, including performance expectancy, effort expectancy, and TTF (Abbad, 2021; Wang & Kartika Sari, 2024). Higher behavioural intention is associated with greater engagement, improved performance, and stronger adoption continuity (Cheng et al., 2020; Thabet et al., 2024). In Thonny-py5mode-supported learning, beginner-friendly features, graphical feedback, extendable libraries, and a low learning curve (with the capacity to support more advanced projects over time) aim to

address both task and personal needs, which should increase students' willingness to adopt and continue using it (Wang & Kartika Sari, 2024).

4. COURSE SETTING

This study evaluated students' perceptions of Thonny-py5mode in an introductory programming offering at an Australian university. The course was a 12-week, first-year-level unit delivered in Trimester 1, 2025, introducing Python to beginners. During the first four weeks of the course, students worked in IDEs such as Jupyter Notebook, PyCharm, and Spyder, producing console-based output only. Instruction during this period focused on foundational programming concepts, culminating in Assessment 1.

Assessment 1, due at the end of Week 4, tested students' understanding of core Python syntax, decision-making, and data structures through text-based output.

Assessment 2, due at the end of Week 8, required students to complete a series of graphical coding challenges using Thonny-py5mode. Weeks 5–8 introduced students to Thonny-py5mode. This second instructional block built on the foundational concepts introduced earlier, extending them into graphical programming tasks using Python and the py5 library. Specifically, the Assessment 2 tasks required students to apply *if–else* statements and *for* and *while* loops in combination with a limited set of approximately 15 py5 drawing functions, with additional marks awarded for clear program structure and effective commenting. Figure 3 presents the six tasks for students to recreate using Python code, drawing on knowledge developed across the course.

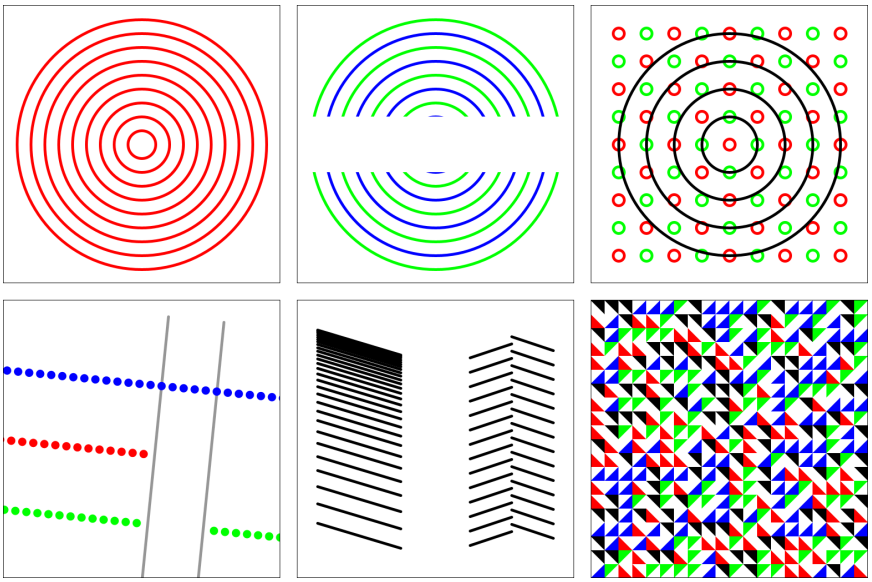


Figure 3. Thonny-py5mode Assessment 2 Challenges

To summarise, the weekly content leading up to Assessment 2 covered: (Week 1) Introduction to Computer Programming and IDEs; (Week 2) Fundamentals of Decision Logic; (Week 3) Intermediate Decision Logic; (Week 4) Lists and Dictionaries; (Week 5) Nested Decision Logic; (Week 6) Simple Loops; (Week 7) Basic Functions; and (Week 8) Intermediate Loops, *with a particular emphasis on the instructional content and learning activities of Weeks 5–8* (which explicitly build on those concepts introduced in Weeks 1–4). Weeks 9–12 content fell outside the scope of this study.

5. METHODOLOGY

Our proposed model (Figure 2) examined key factors shaping novice learners' behavioural intention to learn Python programming using Thonny-py5mode. The methodology employed a quantitative, survey-based approach, selected for its confirmatory and deductive qualities and its typical application in studies that apply TTF and other technology acceptance frameworks (Alyoussef, 2021; Budiartha et al., 2024; Wang & Kartika Sari, 2024). The model involved testing a priori hypotheses and examining theoretically grounded relationships using data from real-world educational settings (Furneaux, 2012; Mosunmola et al., 2018; Park, 2019).

We applied a structured validation approach to develop and validate the measurement instrument: a survey questionnaire comprising seven sections (Table 1) aligned with our model constructs (see Figure 2). We first developed our model through a literature review to identify and define theoretical constructs, refining these through specialist (Python lecturer) input and analysis. Experts in Python teaching and learning reviewed the draft questionnaire to improve its structure and wording. Based on feedback, we amended potentially ambiguous statements to establish the instrument's content validity (Budiartha et al., 2024; Yaakop et al., 2020).

We recruited participants aged 18 years or older from campuses in Adelaide, Brisbane, Melbourne, Sydney, and online. From Week 9, following the submission of Assessment 2, we invited students to complete a voluntary, anonymous questionnaire either online or on paper, with no incentives offered and assurances that participation would not affect their grades. All items used a 5-point Likert scale (1 = strongly disagree to 5 = strongly agree) addressing constructs across the study's extended TTF model. Table 1 outlines the model constructs alongside their corresponding item groupings. To ensure construct–item congruence, each measurement item in Table 1 directly derives from its theoretical construct, aligning with the hypothesised relationships in the extended TTF model.

The researchers conducting the study did not teach or grade the course, and students were aware that the Thonny-py5mode was under evaluation. The online questionnaire closed one week after the 12-week course concluded (it opened in Week 9), and we distributed paper questionnaires for completion up until Week 12 classes. Completion time averaged 10–15 minutes across both in-class and online administrations. All participants provided informed consent, and the study complied with the research ethics guidelines approved by the TUA Human Research Ethics Committee (HREC; Application 0394), classified as Low and Negligible Risk (LNR).

We employed Structural Equation Modelling (SEM), which requires an adequate sample size to ensure consistent and reliable results. Guidelines for SEM recommend that the minimum sample size exceed the number of distinct model parameters, with a common rule of thumb of 5–10 observations per indicator (Hair et al., 2019; Hair et al., 2021). The conceptual model in this study comprised 24 indicators, suggesting an appropriate sample size of 120–240 participants. We collected data from 157 respondents. Data screening procedures addressed missing values, outliers, normality, and multicollinearity (Hair et al., 2019, pp. 62-73, 196-198), resulting in the removal of 14 cases. Table 2 presents an overview of the screened respondent statistics, comprising 143 valid responses, exceeding the minimum (120) recommended for SEM and thereby supporting the reliability of the analyses.

We selected and applied all statistical analysis procedures, including SEM, Confirmatory Factor Analysis (CFA), reliability testing, and goodness-of-fit evaluation, in accordance with established theory for latent variable modelling and prior empirical studies employing the TTF framework (Bere, 2018; Park, 2019).

Construct	Items for Modelling
Task Characteristics (TC) (sec. 3.1)	• Thonny-py5mode supports effective hands-on learning through graphics creation using Python. • Programming graphics using Thonny-py5mode encourages critical thinking and analytical skills. • Overall, Thonny-py5mode is an effective environment for learning programming fundamentals.
Technology Characteristics (TCs) (sec. 3.2)	• Thonny-py5mode features an intuitive, user-friendly interface. • I can complete graphical coding tasks efficiently using Thonny-py5mode. • Thonny-py5mode provides reliable, stable performance during coding sessions.
Personalised Learning (PL) (sec. 3.4)	• Using Thonny-py5mode to code graphical output supports my ability to learn at my own pace. • I can customise the Thonny-py5mode environment to match my learning preferences. • Thonny-py5mode accommodates different learning styles and strategies. • Overall, Thonny-py5mode promotes a personalised learning experience.
Hedonic Motivation (HM) (sec. 3.5)	• Thonny-py5mode allows me to express my creativity through coding. • Learning to code graphical output with Thonny-py5mode is fun and enjoyable. • Thonny-py5mode motivates me to experiment with different coding techniques and ideas. • Overall, coding graphical output with Thonny-py5mode is a positive and enjoyable experience.
Effort Expectancy (EE) (sec. 3.6)	• Thonny-py5mode is easy to install and set up. • Thonny-py5mode is easy to learn for beginners. • Thonny-py5mode makes it easy to debug code. • Overall, Thonny-py5mode is efficient and easy to use.
Task–Technology Fit (TTF) (sec. 3.7)	• Thonny-py5mode makes it easy for me to learn new coding concepts. • Thonny-py5mode provides useful and efficient debugging tools. • Overall, Thonny-py5mode supported the enhancement of my coding skills.
Behavioural Intention (BI) (sec. 3.8)	• I intend to continue using Thonny-py5mode for future projects. • I would recommend Thonny-py5mode to others learning to code. • I expect to use Thonny-py5mode frequently for regular graphics and interactive coding activities.

Table 1. Measurement Items for the Study

Measure	Item	Count	Percentage
Participant Campus	Melbourne	34	23.78%
	Sydney	29	20.28%
	Adelaide	15	10.49%
	Brisbane	13	09.09%
	Online	52	36.36%
Gender	Male	91	63.64%
	Female	52	36.36%
Age	18-22	71	49.65%
	23-27	47	32.87%
	28-32	18	12.59%
	33+	7	04.90%

Table 2. Demographics

6. DATA ANALYSIS AND RESULTS

Prior to SEM estimation, the data were assessed for univariate normality using skewness and kurtosis statistics. Across all measurement items, skewness values ranged from -0.94 to -0.21, while kurtosis values ranged from -0.88 to 1.67. All values fell within acceptable thresholds for SEM (|skewness| < 2; |kurtosis| < 7), indicating no substantial departures from normality. These results support the suitability of the data for subsequent CFA and SEM analyses.

We further examined the dataset for common-method bias using Harman's single-factor test and the marker-variable test (Hair et al., 2021). Harman's single-factor test was conducted using an unrotated exploratory factor analysis including all measurement items. The results indicated that multiple factors emerged, with the largest factor accounting for 38.2% of the total variance, which is below the commonly accepted threshold of 50%. This suggests that no single factor dominated the variance and that common-method bias is unlikely to be a significant concern.

We also performed a marker-variable test to further assess common-method bias. We included a theoretically unrelated marker variable, examining its correlations with the substantive constructs. The average correlation between the marker variable and the main study constructs was r = 0.07, while the average substantive construct correlations exceeded r = 0.40. After partialling out the marker-variable effects, the substantive path coefficients remained stable and statistically significant. These results indicate that method variance is minimal relative to substantive variance. Together, these findings suggest a low likelihood of common method bias in this study.

Following data preparation and bias checks, we applied SEM to test and validate the research model, examining the critical determinants of students' intention to learn Python programming using the Thonny-py5mode coding environment. To empirically test the validity and reliability of the constructs presented in Figure 2, we conducted a CFA in AMOS version 26 using the survey data. Before assessing model fit in AMOS, the measurement instrument was first evaluated for internal consistency and construct validity using Cronbach's alpha (α), Composite Reliability (CR), item reliability (IR), and CFA to ensure the robustness of the measurement model. The goodness-of-fit (GOF) statistics assessed were the likelihood ratio chi-square (χ^2), the ratio of χ^2 to degrees of freedom (χ^2/df), the root mean square error of approximation (RMSEA), and the Comparative Fit Index (CFI). Table 3 reports the results of the measurement model evaluation.

Construct	Item	FL	IR	CR	χ^2/df	p	CFI	RMSEA	α
Recommended value		*≥0.70*	*≥0.50*	*≥0.50*	*≤0.30*	*≥0.05*	*≥0.90*	*≤0.08*	*≥0.70*
Task Characteristics (TC)	TC1	0.84***	0.71	0.93	1.67	0.14	0.98	0.06	0.91
	TC2	0.89***	0.79						
	TC3	0.91***	0.83						
Technology Characteristics (TCs)	TCs1	0.83***	0.70	0.90					0.90
	TCs2	0.79***	0.68						
	TCs3	0.85***	0.73						
Personalised learning (PL)	PL1	0.77***	0.65	0.88					0.86
	PL2	0.83***	0.70						
	PL3	0.86***	0.74						
	PL4	0.86***	0.74						
Hedonic Motivation (HM)	HM1	0.77***	0.75	0.89					0.85
	HM2	0.73***	0.63						
	HM3	0.84***	0.71						
	HM4	0.81***	0.68						
Effort Expectancy (EE)	EE1	0.94***	0.88	0.94					0.92
	EE2	0.92***	0.82						
	EE3	0.88***	0.77						
	EE4	0.91***	0.81						
Task Technology Fit (TTF)	TTF1	0.79***	0.68	0.98					0.86
	TTF2	0.86***	0.74						
	TTF3	0.83***	0.70						
Behavioural Intention (BI)	BI1	0.84***	0.71	0.93					0.91
	BI2	0.94***	0.88						
	BI3	0.92***	0.85						
*Note(s): ***p ≤ 0.001, **p ≤ 0.01, *p ≤ 0.05*									

Table 3. Measurement Model Validation and Reliability Statistics

As shown in Table 3, all measurement items exhibit strong psychometric properties, confirming the adequacy of the measurement model. Standardised factor loadings (FL) range from 0.73 to 0.94 and are all statistically significant ($p \leq 0.001$), exceeding the minimum acceptable threshold of 0.50 and the preferred level of 0.70 (Hair et al., 2019). Item reliability (IR), assessed using squared multiple correlations, exceeds 0.50 for all items, indicating adequate reliability.

Composite Reliability (CR) values for all constructs range from 0.88 to 0.98, well above the recommended threshold, demonstrating high internal consistency and strong convergent validity. Construct reliability was further evaluated using Cronbach's alpha (α), with all coefficients exceeding 0.85 and surpassing the acceptable benchmark of 0.70 (Hair et al., 2019). Collectively, these results provide robust evidence of convergent validity and construct reliability, justifying progression to goodness-of-fit assessment and subsequent structural model analysis.

We assessed the GOF statistics for the final measurement model after conducting validity and reliability tests. The normalised chi-square value (χ^2/df = 1.67) fell below the recommended cut-off of 3.00,

indicating that the model did not significantly differ from the observed data. The RMSEA value (0.06) was below the 0.08 threshold, and the CFI value (0.98) exceeded the 0.90 threshold. Together, these results indicate a good fit between the model and the data, making it suitable for hypothesis testing.

Discriminant validity is supported when the square root of the AVE for each construct is greater than its correlations with other constructs (Hair et al., 2019). Table 4 presents the correlation matrix for the seven constructs. All constructs demonstrated high discriminant validity, with AVE values ranging from 0.77 to 0.85. For example, the AVE for Effort Expectancy (0.85) exceeds its correlations with other constructs, which range from 0.22 to 0.77, indicating strong discriminant validity.

	TC	TCs	PL	HM	EE	TTF	BI
TC	**0.82**						
TCs	0.56	**0.79**					
PL	0.48	0.42	**0.84**				
HM	0.62	0.52	0.47	**0.82**			
EE	0.39	0.49	0.53	0.38	**0.85**		
TTF	0.51	0.44	0.41	0.36	0.28	**0.77**	
BI	0.45	0.38	0.49	0.42	0.41	0.22	**0.81**

Table 4. AVE and Squared Correlation Matrix

We assessed the overall fit of the structural model using multiple fit indices to evaluate the proposed hypotheses. Table 5 summarises the results. The chi-square (χ^2) value normalised by the degrees of freedom (χ^2/df) was 2.34, below the recommended cut-off of 3. The GFI (0.92) and AGFI (0.93) exceeded the recommended threshold of 0.80, while the TLI (0.98) and CFI (1.12) were above the 0.90 threshold (note that AMOS occasionally reports CFI values slightly above 1.0 due to rounding). The RMSEA (0.07) was below the recommended maximum of 0.08. Overall, these results indicate that the structural model provides a good fit to the data (Hair et al., 2019).

Model Fit Indices	Recommended Value	Actual Value
χ2/df	1 < NC < 3	2.34 ($p \leq 0.001$)
RMSEA	<0.08	0.07
GFI	>0.80	0.92
AGFI	>0.80	0.93
TLI	>0.90	0.98
CFI	>0.90	1.12

Table 5. Research Model's Goodness-of-Fit (GOF) Indices

Table 6 presents the results of hypothesis testing for the structural model. The path coefficients (β) and significance levels (p-values) provide statistical support for eight hypotheses (H1, H2, H4, H5, H6, H7, H8, and H9), while there is insufficient support for H3.

Hypothesis		Path		Coefficient (β)	SE	p-value	Model Support
H1	TC	→	HM	0.43	0.065	0.003	Supported
H2	TCs	→	EE	0.18	0.067	0.000	Supported
H3	PL	→	TC	0.02	0.072	0.170	Not Supported
H4	TC	→	TTF	0.56	0.055	0.000	Supported
H5	TCs	→	TTF	0.66	0.049	0.005	Supported
H6	PL	→	TTF	-0.33	0.061	0.003	Supported
H7	HM	→	TTF	0.26	0.054	0.000	Supported
H8	EE	→	TTF	0.41	0.060	0.000	Supported
H9	TTF	→	BI	0.33	0.057	0.000	Supported

Table 6. Structural Model Path Coefficients and Hypothesis Testing Results

Figure 4 presents the model from Figure 2 with the corresponding path coefficients, standard errors, and significance levels for each hypothesised relationship in the extended TTF model.

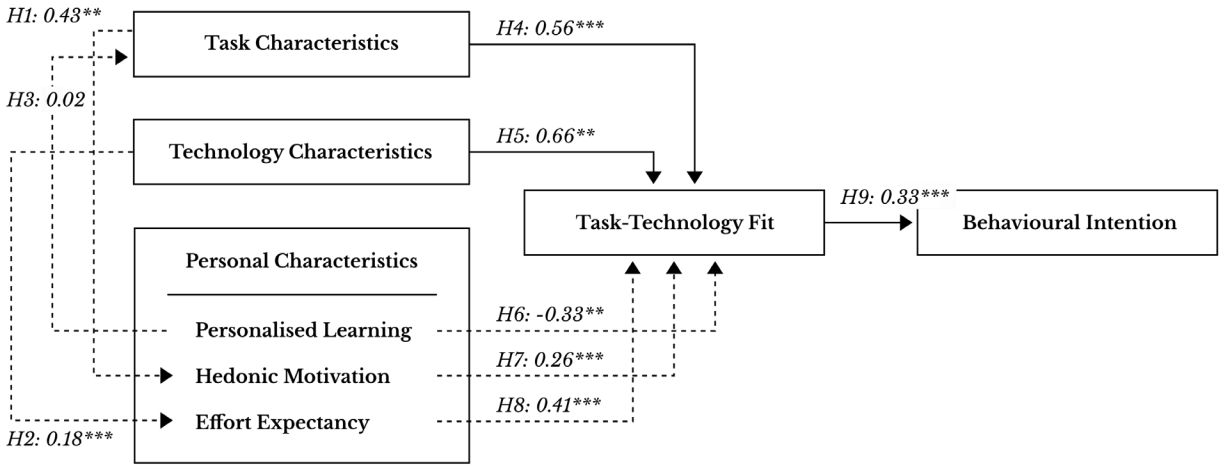


Figure 4. Extended TTF Model With Path Estimates and Significance

The hypothesis-testing results indicated strong support for most of the proposed relationships in the extended TTF model. **H1** – Task characteristics positively influenced hedonic motivation in Thonny-py5mode-supported learning (β = 0.43, p = 0.003), indicating that engaging, well-structured tasks enhanced enjoyment and emotional engagement. **H2** – Technology characteristics positively influenced effort expectancy (β = 0.18, p < 0.001), suggesting that user-friendly features reduced the perceived effort required to use Thonny-py5mode. **H3** – Personalised learning did not significantly influence task characteristics (β = 0.02, p = 0.170), indicating that adaptive or individualised elements did not alter perceptions of the learning tasks. **H4** – Task characteristics positively influenced task–technology fit (β = 0.56, p < 0.001), showing that clarity and relevance shaped perceptions of tool–task alignment. **H5** – Technology characteristics positively influenced task–technology fit (β = 0.66, p = 0.005), underscoring the role of usability and functionality. **H6** – Personalised learning negatively influenced task–technology fit (β = −0.33, p = 0.003), suggesting that greater personalisation sometimes reduced perceived alignment. **H7** – Hedonic motivation positively influenced task–technology fit (β = 0.26, p < 0.001), affirming that emotional engagement improved perceptions of alignment. **H8** – Effort expectancy positively influenced task–

technology fit ($\beta = 0.41$, $p < 0.001$), indicating that ease of use strengthened perceived fit. **H9** – Task–technology fit positively influenced behavioural intention to use Thonny-py5mode ($\beta = 0.33$, $p < 0.001$), showing that students were more likely to adopt the environment when it aligned with their learning needs.

7. DISCUSSION

Our structural model (Figure 4) shows that task characteristics, technology characteristics, personalised learning, hedonic motivation, and effort expectancy all have direct, significant effects on the TTF of Thonny-py5mode. Moreover, this fit has a direct, significant impact on students' adoption of Thonny-py5mode for learning Python across four campuses in Australia and its online cohort. However, personalised learning did not significantly affect technology characteristics in this study.

H1 tested whether task characteristics influence hedonic motivation, demonstrating that the model supports H1 ($\beta = 0.43$, $p = 0.003$). This aligns with previous findings that well-structured, cognitively engaging learning tasks can significantly enhance learners' enjoyment: Škorić et al. (2021) found that aligning programming task demands with technology capabilities improves student satisfaction and continued use intentions; similarly, Melzer (2019) noted that task design (an aspect of task characteristics) can strengthen motivation and engagement in information systems education. In our context, Thonny-py5mode's support for step-by-step debugging, output visualisation, and structured logic-based graphical challenges appears to heighten students' emotional involvement, thereby raising hedonic motivation. These task effects also ripple into effort expectancy perceptions. When learners like using a tool because it structures tasks clearly and responds instantly, they tend to see the tool as less effortful. In this study, Thonny-py5mode matched the structure of students' tasks and provided immediate feedback, increasing intrinsic motivation and creating a more pleasant Python learning experience.

H2 examined the link between technology characteristics and effort expectancy, confirming that Thonny-py5mode significantly influenced the latter for students learning Python programming ($\beta = 0.18$, $p = 0.007$). Prior research finds that usability, clear feedback, and responsive system design reduce cognitive load and increase perceived ease of use. Wang and Kartika Sari (2024) reported that user-friendly interface features and visual feedback mechanisms enhance perceived ease of use in educational technology contexts; Škorić et al. (2021) observed that web-based programming tools offering simplified navigation, real-time debugging, and platform accessibility support higher levels of perceived task–technology fit and reduce effort required. Thonny-py5mode's intuitive layout, error reporting, and graphical output lower the cognitive and physical effort students expend during learning. This pattern suggests that effort expectancy is shaped by how well the system aligns with novices' cognitive needs. When friction drops, learners can focus on grasping Python concepts rather than wrestling with the environment.

H3 tested whether personalised learning shapes task characteristics. Contrary to some prior work, our model did not support H3 ($\beta = 0.02$, $p = 0.170$). Melzer (2019) reported that personalisation, when grounded in cognitive fit theory, can improve learners' perceptions of task clarity in information systems education. However, our findings suggest that personalisation alone does not automatically yield clearer or better-structured tasks. To influence task characteristics, system designers likely need to embed personalisation into the tasks' logic and scaffolding, not just into interface-level features.

H4 concerned the impact of task characteristics on task–technology fit, demonstrating a significant, positive effect ($\beta = 0.56$, $p < 0.001$). This positive relationship suggests that Thonny-py5mode's supportive interface design, such as its syntax alerts, integrated visual feedback, and simplified debugging, may contribute to learners' perception of task–technology alignment. Prior research confirms that, when tasks are clearly defined and supported by interactive affordances such as real-time feedback and visualisation, students perceive a stronger TTF (Liu et al., 2023). Yaakop et al. (2020) found that collaborative and well-structured tasks in cloud-based learning systems improved perceived alignment between tasks and technological tools.

H5 tested whether technology characteristics directly influence task–technology fit, showing a strong positive relationship ($\beta = 0.66$, $p < 0.005$). Research from Thabet et al. (2024) shows that system quality elements—particularly intuitive interfaces, real-time visual output, and error-handling assistance—improve

perceptions of how well a technology matches task demands; similarly, Wang and Kartika Sari (2024) observed that platform responsiveness and adaptability enhance user engagement and strengthen TTF perceptions. In this study, the approachable Thonny-py5mode interface, convenient installation process, graphical output, and shell panel support closely match the requirements of creative coding tasks. Through this alignment, students can focus on problem-solving rather than wrestling with the environment, thereby reinforcing perceptions of strong TTF.

H6 indicates that personalised learning had a significant negative effect on task–technology fit ($\beta = -0.33$, $p = 0.003$), suggesting that greater personalisation was associated with lower perceived alignment between Thonny-py5mode and the programming tasks. This aligns with H3's finding that personalisation did not significantly improve task characteristics, implying limited integration with the task structure. In contrast, Melzer (2019) found that tailoring environments to individual preferences can strengthen perceived fit; Ishaq et al. (2024) reported that adaptive difficulty, personalised feedback, and real-time code visualisation can enhance alignment in programming contexts. One possible explanation is that aspects of Thonny-py5mode that could support personalisation, such as immediate visual feedback and the creative flexibility of py5 commands and graphical output, were not tailored to individual learners or sufficiently integrated into the assessment scaffolding. This made them seem loosely connected to the programming tasks and fostered a perception of mismatch rather than synergy.

H7 addresses hedonic motivation's role in task–technology fit and indicates a significant, positive relationship ($\beta = 0.26$, $p < 0.001$). Škorić et al. (2021) found that, when students perceive interactive programming environments as enjoyable, their assessment of TTF improves; Ishaq et al. (2024) similarly reported that adaptive difficulty, personalised feedback, and real-time code visualisation can heighten both enjoyment and perceived alignment in programming contexts. Thonny-py5mode's graphics-oriented output provides a form of immediate visual feedback (alongside shell output), and its simplified interface likely makes programming less intimidating for Python novices.

H8 tested the effect of effort expectancy on task–technology fit, finding a significant, positive relationship ($\beta = 0.41$, $p < 0.001$). Prior research supports this link between perceived ease of use and stronger perceptions of task–technology alignment. Lim and Lee (2021) found that intuitive interfaces and reduced complexity enhance users' sense of fit with learning tasks; Thabet et al. (2024) reported that clear navigation structures, immediate feedback, and low cognitive load significantly improve the perceived match between system capabilities and intended tasks. Thonny-py5mode's clean layout, integrated debugging, and graphical output likely reduced effort, reinforcing perceptions of fit with the graphical programming tasks.

H9 links task–technology fit to behavioural intention. The effect is significant and positive ($\beta = 0.33$, $p < 0.001$). DeLone and McLean's (2003) IS success model emphasises the role of system–task alignment in driving continued use; Škorić et al. (2021) confirmed that when students perceive web-based programming tools as fitting well with learning activities, they show stronger continuance intentions.

Taken together, these results highlight an integrated picture of adoption. Task–technology fit is a central driver of behavioural intention, shaped by clear, well-scaffolded tasks (H1, H4), strong technology characteristics (H2, H5), and user-centric elements that make the experience both enjoyable (H7) and easy (H8). In Thonny-py5mode, features such as syntax alerts, intuitive debugging, and integrated graphical output closely align with the demands of graphical Python tasks. Personalised learning, however, shows a more complex pattern: it does not significantly alter task characteristics (H3) and, in this study, was associated with lower TTF (H6). This suggests that when adaptive elements, such as immediate visual feedback or the creative possibilities offered by py5 commands, are not tightly embedded in task scaffolding, they may reduce rather than enhance perceived alignment.

For practice, three implications stand out: (1) design tasks that are structured, cognitively engaging, and tightly mapped to learning outcomes, while ensuring the environment provides immediate, meaningful feedback; (2) adopt environments with novice-friendly interfaces, visual feedback, and streamlined debugging features that reliably lower effort, raise enjoyment, and strengthen TTF; and (3) treat personalisation with care by ensuring that task scaffolding and learning pathways closely integrate with adaptive pacing and feedback, rather than layering these superficially at the interface level.

We do, however, note a limitation of this study: the sample consisted primarily of first-year students in an introductory Python course. While appropriate for examining early adoption, prior programming experience may moderate how learners perceive fit and adoption. Future work will therefore include more cross-cohort and experienced-student groups to evaluate the influence of previous programming experience and IDE exposure.

In the specific case of Thonny-py5mode, the convergence of usability, visual immediacy, and supportive debugging appears to deliver a strong perceived fit to core Python programming tasks. Students who experienced this fit reported greater enjoyment, lower perceived effort, and stronger intentions to continue using the environment. The overall pattern suggests that Processing-like, graphically oriented Python environments such as Thonny-py5mode can meaningfully enhance early programming experiences when they balance clear task design, robust interactive affordances, and thoughtfully embedded personalisation.

Future research should adopt a longitudinal design to examine how task–technology fit and behavioural intention evolve as students' progress from introductory to advanced programming courses, where changing task complexity, expertise, and learner expectations may alter the features required to sustain perceived fit and continued use.

In addition to longitudinal designs, future research would benefit from adopting mixed methods approaches that incorporate qualitative data to complement SEM-based findings. While this study identifies statistically significant relationships among task characteristics, technology characteristics, and learner personal characteristics, qualitative methods such as interviews, focus groups, or reflective learning journals could provide deeper insight into how students interpret and experience these relationships in practice. For example, qualitative data could help explain why personalised learning exhibited a negative relationship with task–technology fit, uncovering learners' perceptions of misalignment between adaptive features and assessment scaffolding. Mixed methods designs would also allow researchers to capture contextual factors such as learner expectations, prior programming experience, and affective responses that are not fully observable through survey instruments alone. By integrating qualitative insights with quantitative modelling, future studies could develop a more nuanced understanding of user experience, sense-making processes, and pedagogical implications associated with creative-coding environments such as Thonny-py5mode.

8. RESEARCH IMPLICATIONS

The study offers both theoretical and practical contributions for the use of Thonny-py5mode. Theoretically, it extends TTF theory by incorporating three personal characteristics: personalised learning, hedonic motivation, and effort expectancy. While traditional TTF frameworks focus on aligning task needs and technology characteristics (Furneaux, 2012), our model posits that learners' experiences and motivations also mediate this relationship. Testing hypotheses 1 through 9, we confirm that perceived fit depends on both system functionality and users' enjoyment, learning needs, and intention to use the system, echoing prior work demonstrating the joint influence of system quality and user factors on TTF (Škorić et al., 2021; Thabet et al., 2024), expanding TTF to include emotional and motivational dimensions alongside functional alignment. The insignificant effect between personalised learning and technology characteristics further suggests that adaptive learning systems must consider not only the quality of system features but also the user's context and capability, consistent with insights from Melzer (2019) on the role of cognitive fit in personalised learning design. Design implications resonate with findings that effective curriculum transformation requires embedding authentic uses of technology in programme design (Rambe, 2016). That sustained self-efficacy support through hands-on, scaffolded training enables educators to adopt more constructive, knowledge-building uses of ICT in classrooms (Mlambo et al., 2020).

Beyond its theoretical extension of the TTF framework, this study contributes methodologically by providing a rigorously validated measurement model with robust reliability and construct validity for learner-centred constructs in programming environments for graphical output.

Practically, the findings suggest selecting or designing coding environments like Thonny-py5mode for their ability to support structured learning tasks while remaining stimulating and user-friendly. In Thonny-py5mode, features such as graphical output via py5 sketches, seamless integration between the sketch window and Thonny panels, and a simplified, non-command-line interface that avoids package management reduce cognitive load and boost satisfaction (Lim & Lee, 2021; Thabet et al., 2024). These elements align closely with our results on task–technology fit, as the environment's layout and immediacy in generating graphics help beginners stay oriented and engaged. The strong link between hedonic motivation and TTF is illustrated by Thonny-py5mode's ability to make coding sessions more enjoyable through real-time graphical results, encouraging sustained use (Škorić et al., 2021; Yaakop et al., 2020). For educators, these findings can inform the adaptation of existing IDEs or the design of new ones tailored to specific Python multimedia libraries, without sacrificing ease of use. By pairing beginner-friendly tools with scaffolded coding tasks and challenges, educators can help novices build confidence and skills. Finally, the significant influence of TTF on behavioural intention shows that, when students feel Thonny-py5mode fits their needs, they are more inclined to continue using it—underscoring the importance of aligning the tool's features with both pedagogical goals and learner characteristics (DeLone & McLean, 2003).

This study further contributes by demonstrating that personalisation implemented at the user-experience level alone is insufficient to enhance learning, underscoring the need for pedagogically embedded and task-aligned personalisation within extended TTF models.

9. CONCLUSION

This study applied an extended Task–Technology Fit framework to examine factors shaping university students' intention to use Thonny-py5mode for learning Python. The results show that task characteristics, technology characteristics, hedonic motivation, effort expectancy, and personalised learning all influenced perceived fit, with TTF in turn driving behavioural intention. While personalisation did not significantly shape task characteristics and, in this case, was associated with lower TTF, the combination of structured tasks, user-friendly design, and an engaging graphical approach created strong perceived alignment between Thonny-py5mode and learners' needs.

For practice, our findings underscore the value of beginner-friendly IDEs that integrate graphical output and beginner-friendly design, complemented with scaffolded tasks to reduce cognitive load and sustain engagement. We demonstrated this by extending TTF theory to integrate motivational constructs alongside functional alignment in predicting technology adoption in programming education. In Thonny-py5mode, the balance between usability and a workflow oriented toward graphical output reduced effort and encouraged continued use. Future research should examine how personalisation can be more tightly embedded in task scaffolding and how contextual factors, such as prior programming experience and educator support, influence adoption outcomes.

10. AI USE DISCLOSURE

During the preparation of this work, the authors used Grammarly and GPT-4o as writing assistance tools to improve clarity and presentation. All content was subsequently reviewed and revised by the authors, who assume full responsibility for the publication's final content and conclusions.

11. REFERENCES

Abbad, M. M. M. (2021). Using the UTAUT Model to Understand Students' Usage of E-Learning Systems in Developing Countries. Education and Information Technologies, 26(6), 7205-7224. https://doi.org/10.1007/s10639-021-10573-5

Alyoussef, I. Y. (2021). Massive Open Online Course (MOOCs) Acceptance: The Role of Task-Technology Fit (TTF) for Higher Education Sustainability. *Sustainability*, 13(13), 7374. https://doi.org/10.3390/su13137374

Annamaa, A. (2024). Thonny, Python IDE for Beginners. In Thonny. https://thonny.org

Ayasrah, F. T. M. (2020). Exploring E-Learning Readiness as Mediating Between Trust, Hedonic Motivation, Students' Expectation, and Intention to Use Tech. in Taibah University. *Journal of Education & Social Policy*, 7(1). https://doi.org/10.30845/jesp.v7n1p12

Bakar, M. A., Mukhtar, M., & Khalid, F. (2020). The Effect of Turtle Graphics Approach on Students' Motivation to Learn Programming: A Case Study in a Malaysian University. *International Journal of Information and Education Technology*, 10(4), 290-297. https://doi.org/10.18178/ijiet.2020.10.4.1378

Bere, A. (2018). Applying an Extended Task-Technology Fit for Establishing Determinants of Mobile Learning: An Instant Messaging Initiative. *Journal of Information Systems Education*, 29(4), 239-252. https://aisel.aisnet.org/jise/vol29/iss4/4

Bere, A., & Rambe, P. (2016). An Empirical Analysis of the Determinants of Mobile Instant Messaging Appropriation in University Learning. *Journal of Computing in Higher Education*, *28*(2), 172-198. https://doi.org/10.1007/s12528-016-9112-2

Budiartha, C. I. W. E., Redi, A. A. N. P., Halim, G. R., Rachmah, A., & Grahani, B. P. (2024). Using Task-Technology Fit to Analyze the Adoption of Google Workspace for Education in Teaching and Learning Process. *The Asian Conference on Education & International Development 2024 Official Conference Proceedings* (pp. 873-882). https://doi.org/10.22492/issn.2189-101X.2024.69

Cheng, Y., Sharma, S., Sharma, P., & Kulathunga, K. (2020). Role of Personalization in Continuous Use Intention of Mobile News Apps in India: Extending the UTAUT2 Model. *Information*, 11(1), 33. https://doi.org/10.3390/info11010033

Chiodini, L., Sorva, J., Hellas, A., Seppälä, O., & Hauswirth, M. (2025). Two Approaches for Programming Education in the Domain of Graphics: An Experiment. *The Art, Science, and Engineering of Programming*, 10(1), 14. https://doi.org/10.22152/programming-journal.org/2025/10/14

DeLone, W., & McLean, E. (2003). The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*, 19(4), 9-30. https://doi.org/10.1080/07421222.2003.11045748

Feinburg, J. (2023). Python Mode for Processing. In GitHub. https://github.com/jdf/processing.py

Funk, M., & Zhang, Y. (2024). *Coding Art: A Guide to Unlocking Your Creativity With the Processing Language and p5.js in Four Simple Steps* (2nd ed.). APress. https://doi.org/10.1007/978-1-4842-9780-3

Furneaux, B. (2012). Task-Technology Fit Theory: A Survey and Synopsis of the Literature. In Y. K. Dwivedi, M. R. Wade, & S. L. Schneberger (Eds.), *Information Systems Theory* (Vol. 28, pp. 87-106). Springer New York. https://doi.org/10.1007/978-1-4419-6108-2_5

Goodhue, D. L., & Thompson, R. L. (1995). Task-Technology Fit and Individual Performance. *MIS Quarterly*, 19(2), 213-236. https://doi.org/10.2307/249689

Guo, P. (2021). Ten Million Users and Ten Years Later: Python Tutor's Design Guidelines for Building Scalable and Sustainable Research Software in Academia. *The 34th Annual ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/3472749.3474819

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate Data Analysis* (8th ed.). Cengage.

Hair, J. F., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Danks, N. P., & Ray, S. (2021). An Introduction to Structural Equation Modeling. In J. F. Hair Jr., G. T. M. Hult, C. M. Ringle, M. Sarstedt, N. P. Danks, & S. Ray (Eds.), *Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R: A Workbook* (pp. 1-29). Springer International Publishing. https://doi.org/10.1007/978-3-030-80519-7_1

Hidayat, D., Pangaribuan, C. H., Putra, O. P. B., & Irawan, I. (2021). Contemporary Studies of Task-Technology Fit: A Review of the Literature. *2021 International Conference on Information Management and Technology* (ICIMTech) (pp. 309-313). https://doi.org/10.1109/ICIMTech53080.2021.9535028

Hunt, J. (2019). Python Turtle Graphics. In *Advanced Guide to Python 3 Programming* (pp. 13-21). Springer International Publishing. https://doi.org/10.1007/978-3-030-25943-3_3

Hussain, K., & Hussain, F. (2024). *Crafting Games With Python & Pygame*. Sonar Publishing.

Ishaq, K., Alvi, A., Haq, M. I. U., Rosdi, F., Choudhry, A. N., Anjum, A., & Khan, F. A. (2024). Level Up Your Coding: A Systematic Review of Personalized, Cognitive, and Gamified Learning in Programming Education. *PeerJ Computer Science*, 10, e2310. https://doi.org/10.7717/peerj-cs.2310

Lim, T. L., & Lee, A. S. H. (2021). Extended TAM and TTF Model: A Framework for the 21st Century Teaching and Learning. *2021 International Conference on Computer & Information Sciences* (ICCOINS) (pp. 334-339). https://doi.org/10.1109/ICCOINS49721.2021.9497216

Liu, K., Yao, J., Tao, D., & Yang, T. (2023). Influence of Individual-Technology-Task-Environment Fit on University Student Online Learning Performance: The Mediating Role of Behavioral, Emotional, and Cognitive Engagement. *Education and Information Technologies*, 28(12), 15949-15968. https://doi.org/10.1007/s10639-023-11833-2

Melzer, P. (2019). A Conceptual Framework for Task and Tool Personalisation in IS Education. In *A Conceptual Framework for Personalised Learning* (pp. 47-76). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-23095-1_3

Mlambo, S., Rambe, P., & Schlebusch, L. (2020). Effects of Gauteng Province's Educators' ICT Self-Efficacy on Their Pedagogical Use of ICTS in Classrooms. *Heliyon*, 6(4), e03730. doi:10.1016/j.heliyon.2020.e03730

Mosunmola, A., Mayowa, A., Okuboyejo, S., & Adeniji, C. (2018). Adoption and Use of Mobile Learning in Higher Education: The UTAUT Model. *Proceedings of the 9th International Conference on E-Education, E-Business, E-Management and E-Learning* (20-25). https://doi.org/10.1145/3183586.3183595

Park, C. (2019). Exploring a New Determinant of Task Technology Fit: Content Characteristics. *Journal of International Technology and Information Management*, 27(3), 100-118. https://doi.org/10.58729/1941-6679.1385

Rambe, P. (2016). The Role of Educational Technology in Design and Delivery of Curricula Programmes: A Case of STEPS at a University of Technology. *The African Journal of Information Systems*, 8(2), 86-113.

Shahzad, M. F., Xu, S., Hanif, S., & Akram, S. (2025). What Factors Affect the Intention to Use Information and Communication Technology? Perspectives of Chinese University Students. *Humanities and Social Sciences Communications*, 12(1), 847. https://doi.org/10.1057/s41599-025-05156-5

Shiffman, D. (2024). *The Nature of Code: Simulating Natural Systems With JavaScript*. No Starch Press.

Škorić, I., Orehovački, T., & Ivašić-Kos, M. (2021). Task-Technology Fit and Continuance of Use of Web-Based Programming Tool: A Pilot Study. In W. Karwowski, T. Ahram, D. Etinger, N. Tanković, & R. Taiar (Eds.), *Human Systems Engineering and Design III* (Vol. 1269, pp. 57-62). Springer International Publishing. https://doi.org/10.1007/978-3-030-58282-1_10

Tamilmani, K., Rana, N. P., Wamba, S. F., & Dwivedi, R. (2021). The Extended Unified Theory of Acceptance and Use of Technology (UTAUT2): A Systematic Literature Review and Theory Evaluation. *International Journal of Information Management*, 57, 102269. https://doi.org/10.1016/j.ijinfomgt.2020.102269

Thabet, Z., Albashtawi, S., Ansari, H., Al-Emran, M., Al-Sharafi, M. A., & AlQudah, A. A. (2024). Exploring the Factors Affecting Telemedicine Adoption by Integrating UTAUT2 and IS Success Model: A Hybrid SEM–ANN Approach. *IEEE Transactions on Engineering Management*, 71, 8938-8950. https://doi.org/10.1109/TEM.2023.3296132

Venkatesh, V., Morris, M., Davis, G., & Davis, F. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3), 425-478. https://doi.org/10.2307/30036540

Wang, W.-T., & Kartika Sari, M. (2024). Examining the Effect of the Task-Technology Fit of Game Mechanisms on Learning Outcomes in Online Gamification Platforms. *Journal of Educational Computing Research*, 61(8), 1568-1595. https://doi.org/10.1177/07356331231187285

Xu, J., & Frydenberg, M. (2021). Python Programming in an IS Curriculum: Perceived Relevance and Outcomes. *Information Systems Education Journal*, 19(4), 37-54. https://doi.org/10.62273/SINY5870

Yaakop, A. Y., Mahadi, N., Ariffin, Z. Z., Abu Hasan, Z. R., & Harun, M. (2020). Examining Students' Continuance Usage Intention for Web-Based Educational Tools: A Developed Integrated Structural Model Approach. *Asian Academy of Management Journal*, 25(1). https://doi.org/10.21315/aamj2020.25.1.2

Tzeng, S.-Y., Lin, K.-Y., & Lee, C.-Y. (2022). Predicting College Students' Adoption of Technology for Self-Directed Learning: A Model Based on the Theory of Planned Behavior with Self-Evaluation as an Intermediate Variable. *Frontiers in Psychology*, 13, 865803. https://doi.org/10.3389/fpsyg.2022.865803

AUTHOR BIOGRAPHIES



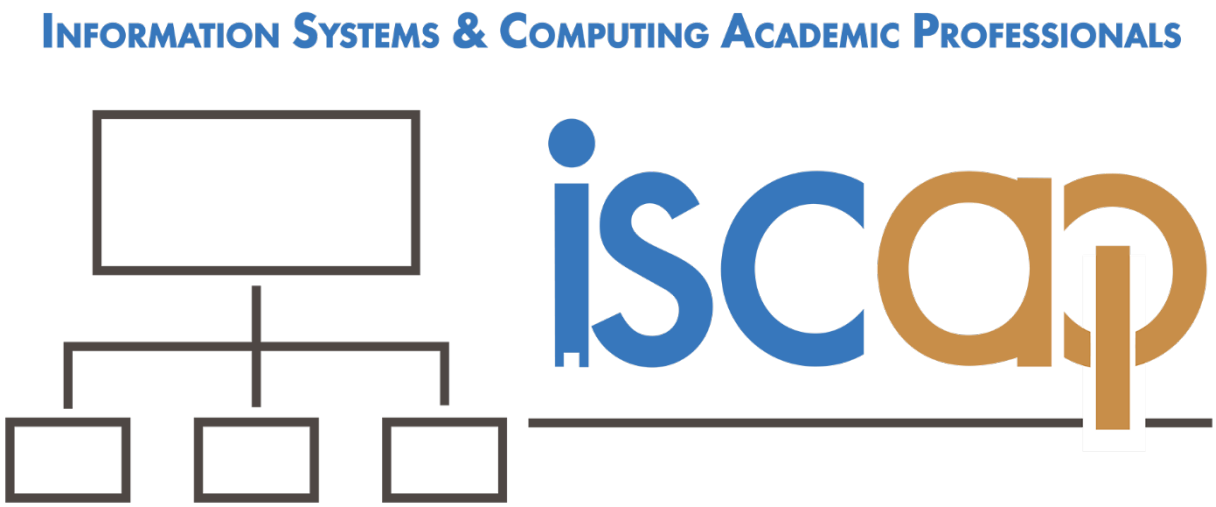
Tristan Bunn is a Creative Technologies Lecturer at Torrens University (Sydney), specialising in UX, web, and interactive media. His practice spans graphic design, interaction design, and multimedia development, with a current research focus on creative applications of machine learning and various emerging media.



Aaron Bere is an Associate Professor and an academic leader in Information Technology and Software Engineering (AI) at Torrens University Australia. He is an active researcher and HDR supervisor with a strong interest in digital pedagogy and emerging technologies.



Michelle Douglas is an Associate Professor and Associate Dean in the School of Design at RMIT University, Australia. She is an expert in design and creative technology education, with extensive experience exploring how design pedagogy shapes students' capacity to engage with emerging creative challenges. Her work examines design culture, place-based behaviours and attitudes, and the integration of emerging technologies, regenerative principles and systems thinking across industrial and interaction design programs. An interdisciplinary academic spanning user-centred and more-than-human approaches, Michelle has held senior leadership roles in the UK and Australia, leading program transformation, research development, and higher degree supervision. Her scholarship contributes to international debates on design pedagogy, sustainability transitions, and responsible, systems-oriented innovation.



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2026 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, ***Journal of Information Systems Education***, editor@jise.org.

ISSN: 2574-3872 (Online) 1055-3096 (Print)