Success Factors of Automated Programming Assessment Systems in Different Learning Environments

Clemens Sauerwein
University of Innsbruck,
Innsbruck, Austria
Clemens.Sauerwein@uibk.ac.at

Tobias Antensteiner
University of Innsbruck,
Innsbruck, Austria
Tobias.Antensteiner@uibk.ac.at

Peter Salhofer
FH Joanneum,
Graz, Austria
Peter.Salhofer@fh-joanneum.at

Ruth Breu
University of Innsbruck,
Innsbruck, Austria
Ruth.Breu@uibk.ac.at

Abstract

In recent years, the demand for programming skills has surged due to the digitization of the economy and society. Consequently, a limited number of lecturers face an increasing number of students from diverse disciplines in programming education. To address this challenge, there is a need for technical solutions that assist teachers in managing programming tasks, automating student solution evaluation, and facilitating objective and standardized feedback. Automated Programming Assessment Systems (APASs) have emerged as a suitable solution to meet these requirements and handle the growing demand for programming education. However, there is a limited understanding of the success factors specific to APASs, as they have primarily been examined in terms of a single system. To address this gap, we conducted a replication study combining data from two distinct learning environments employing different APASs. Our findings are consistent with the results of previous studies, revealing some common success factors of APASs, thereby facilitating their generalization. Nevertheless, we also identified differences that warrant further investigation and analysis.

Keywords: Automated Programmings Assessment Systems, Success Factors, Replication Study

1. Introduction

The rapid growth of the digital economy and the widespread adoption of technology in various sectors have led to a substantial rise in the demand for programming education in universities and colleges. In response to this trend, programming skills, once predominantly valued mainly in the natural sciences, have become indispensable in almost all scientific fields. Consequently, the need for effective programming instruction has expanded, catering to a diverse group of students with varying levels of prior knowledge. However, meeting this escalating demand presents significant challenges due to the scarcity of qualified instructors and the constant evolution of technology, coupled with the rapid pace of advancements in learning technologies (Mekterovic & Brkic, 2017).

To overcome these challenges, the utilization of automated programming assessment systems (APASs) has emerged as a promising solution (Mekterovic & Brkic, 2017). They are purposefully designed to achieve several key objectives, including the efficient management of programming assignments, automated testing of student solutions, and the promotion of objective and standardized feedback (Keuning et al., 2018; Mekterović et al., 2020).

While existing research has focused on the impact of the use of APASs (Barra et al., 2020; Gordillo, 2019; Restrepo-Calle et al., 2019) and the users' experiences (Sauerwein, Priller, et al., 2023), there has been little work to date on the success factors of APASs. A first attempt in this direction was made by Sauerwein, Antensteiner, et al. (2023), who devised a first success model for APASs. However, the majority of existing studies concentrate on a particular APASs utilized in a specific learning environment, implying that research findings can only be generalized to a limited extent (Ullah et al., 2018). Consequently, Sauerwein, Antensteiner's, et al. (2023) model is also limited in its applicability, as it is restricted to the APAS ArTEMiS (Krusche & Seitz, 2018) and was only used in distance learning due to the COVID-19 pandemic (Sauerwein, Antensteiner, et al., 2023).

URI: https://hdl.handle.net/10125/106387
978-0-9981331-7-1

Moreover, the authors emphasize that it would be of particular interest to examine their devised success model in the context of other APASs or learning environments.

To bridge this gap, research at hand presents a replication study that combines study data from two distinct learning environments, namely distance learning and face-to-face teaching, each using a different APAS. In doing so, we employed Sauerwein, Antensteiner's, et al. (2023) proposed success model as well as devised survey instrument to compare their survey results with our own. In contrast to their work, our study did not analyze the role of mediators intervening between technology-, system- or self-related factors, respectively, and the users' satisfaction with APASs (Sauerwein, Antensteiner, et al., 2023). Instead, we aimed to integrate the findings to develop a comprehensive and viable success model for APASs. We found several common success factors supporting a generalization of the model proposed by Sauerwein, Antensteiner, et al. (2023). However, we also identified differences that warrant further investigation and analysis.

The remainder of this paper is structured as follows: Section 2 provides background information on APASs in general and Sauerwein, Antensteiner's, et al. (2023) success model. Section 3 provides an overview of related work and highlights the relevance of this research. Section 4 outlines the research methodology by presenting two different learning environments used in this study. Section 5 presents the results of the multigroup analysis conducted. In section 6, the results are discussed by presenting common success factors as well as their differences across the two employed learning environments. Finally, Section 7 concludes our research and provides an outlook on future work.

2. Background Information

In this section, we define the core functionalities of an APAS (cf. Section 2.1) and introduce the success model for APASs formulated by Sauerwein, Antensteiner, et al. (2023) (cf. Section 2.2).

2.1. Automated Programming Assessment Systems

The main goals of APASs include increasing student motivation, monitoring learning progress, improving the quality of teaching and student solutions, reducing barriers for novice programmers, promoting objective and standardized feedback, and reducing dropout rates (Keuning et al., 2018; Mekterović et al., 2020). Essentially, these platforms facilitate communication with teachers, e.g., by providing exercises and efficient processing of submitted solutions. Students receive support through the access to programming tasks, pre-defined code snippets and the submission of their solutions. Lecturers can assess the submitted programs using provided and self-defined checks and evaluations, e.g., by running test cases, examining the solution structure and performing plagiarism checks, to obtain useful information about their students' performance. Moreover, students can submit their solutions multiple times and learn from the assessments and automatically generated feedback, they gain a comprehensive overview of their performance during the course (Mekterović et al., 2020).

2.2. Success Model

The success model for APASs used as a formative assessment tool proposed by Sauerwein, Antensteiner, et al. (2023) draws on the enhanced model of IS success by DeLone and McLean (2003) as well as Zhang and Dang's (2020) blended learning success model. Therefore, their success model encompasses eight constructs, as presented in Table 1, including three exogenous constructs, *MOT*, *INQ*, *SYQ*, *SEQ*, three mediator constructs, *LEC*, *BLF*, *TTF*, and one endogenous construct, *SAT* (cf. Figure 1).

In particular, Sauerwein, Antensteiner's, et al. (2023) model suggests that *MOT*, *INQ*, *SYQ*, and *SEQ*, respectively, have an indirect effect on *SAT* through *LEC*, *BLF*, and *TTF*.

3. Related Work

Recent research in the field of programming education has placed considerable emphasis on providing support for both lecturers and students in introductory programming (Luxton-Reilly et al., 2018). This line of investigation has explored the potential benefits of automated assessment tools and highlighted their crucial role in programming courses (R. S. Pettit et al., 2015). This has been motivated by a recognition of the challenges faced by learners in acquiring programming skills (Gomes & Mendes, 2007). Accordingly, various tools (Souza et al., 2016) with different strengths and weaknesses have been developed in this regard (Ullah et al., 2018).

Among others, APASs (cf. Section 2.1) are a promising solution to address these hurdles (Mekterovic & Brkic, 2017). Moreover, based on a systematic literature review, R. S. Pettit et al. (2015) showed that an APAS can dramatically reduce teachers' workload while improving students' learning experience. Therefore, it is reasonable that different APAS have emerged over the

Construct	Description
MOT	Motivation describes the APAS's incentives, encouraging the user to engage in learning with the system.
INQ	Information Quality describes the desirable quality characteristics of the APAS's output or content.
SYQ	System Quality describes the desirable quality characteristics of the APAS itself.
SEQ	Service Quality describes the quality of the support provided by the APAS's provider.
LEC	Learning Climate describes the user's enjoyment and pleasure when using the platform, as well as the platform's ability to motivate learning.
BLF	Blended Learning Flexibility describes the degree the APAS offers the freedom to learn at one's own pace, with effective preparation and feedback.
TTF	Task-Technology Fit describes the degree to which the services of an APAS meet the user's learning goals and requirements.
SAT	Satisfaction describes the users' level of satisfaction with the APAS.

Table 1. Constructs of the success model and corresponding descriptions (Sauerwein, Antensteiner, et al., 2023)

past few years (English & English, 2015; Keuning et al., 2018; Krusche & Seitz, 2018; Mekterović et al., 2020). Accordingly, several empirical investigations regarding user experience (Sauerwein, Priller, et al., 2023), user perception (Barra et al., 2020; Gordillo, 2019), improvements in learning programming (R. Pettit et al., 2015) and effects on the learning environment (R. S. Pettit et al., 2015) have been conducted in the context of APASs.

While APASs are seen as information systems (Queiros et al., 2020) and success models for the effective management of information systems are very common (DeLone & McLean, 2003), only a few researchers to date have looked at the success factors of APASs. In this regard, Zhang and Dang (2020) introduced a blended learning success model for APASs, which are used as a formative assessment tool (cf. Section 2.2).

As stated in Section 1, the suitability of the model established by Sauerwein, Antensteiner, et al. (2023) is limited as it only concentrates on distance learning and is restricted to the APAS ArTEMiS (Krusche & Seitz, 2018). Accordingly, we aim to complement their research by conducting a replication study to gain more comprehensive and generalizable insights into the determinants of APASs' success.

4. Applied Research Methodology

To replicate and expand upon the research conducted by Sauerwein, Antensteiner, et al. (2023), we employed their research model (cf. Section 2.2) to define our two hypotheses (cf. Section 4.1), utilized the same questionnaire (cf. Section 4.2), made use of their collected survey data (cf. Section 4.3), and collected supplementary data from students working in a distinct learning environment and using a different APAS (cf. Section 4.4). Subsequently, we conducted a multigroup analysis, in order to (*i*) examine the discrepancies stemming from the two contrasting learning environments and different APASs used, and to (*ii*) identify overarching success factors and investigate their relationships (cf. Section 5).

4.1. Hypotheses

In order to empirically validate Sauerwein, Antensteiner's, et al. (2023) hypothesized model (cf. Section 5.1), we transformed it into a structural equation model, as depicted in Figure 1 and Figure 2, where the two hypotheses we further investigated are represented by arrows connecting the respective constructs denoted by ellipses. In addition, Figure 1 presents the full set of data, i.e., the data from Sauerwein, Antensteiner, et al. (2023) and our replication study, and Figure 2 compares these two data sets individually.

Accordingly, we (*i*) hypotesize that *MOT*, *INQ*, *SYQ* and *SEQ* each have a significant positive association with *LEC*, *BLF*, and *TTF*, respectively, as well as *LEC*, *BLF* and *TTF* each have a significant positive association with *SAT*.

Furthermore, following Henseler's (2007) non-parametric multigroup analysis, we (*ii*) hypothesize, that the path coefficients between the two groups of students having used the two distinct learning environments and different APASs, i.e, the ArTEMiS Environment (cf. section 4.3) on the one hand and the Jupyter Notebook Environment (cf. section 4.4) on the other hand are not significantly different.

4.2. Survey Instrument

To collect data for the statistical analysis to attain a more comprehensive and generalizable understanding of the drivers of APASs' success, we employed a self-administered online questionnaire through LimeSurvey. Therefore, we reused the 33 survey items provided by Sauerwein, Antensteiner, et al. (2023), as listed in Table 2.

Consistent with the original study (Sauerwein, Antensteiner, et al., 2023), all questions in the questionnaire were rated on a five-point Likert scale, ranging from 1 for "strongly disagree" to 5 for "strongly agree". Additionally, demographic information was gathered from participants, including gender, (academic) degree, prior knowledge of programming, and duration of studies to date.

4.3. Data Collection ArTEMiS Environment

As reported by Sauerwein, Antensteiner, et al. (2023), their survey was conducted in a university learning environment during introductory programming courses at the end of the winter semester 2020/21 as well as winter semester 2021/22. Morepver, the APAS ArTEMiS (Krusche & Seitz, 2018) was chosen for these courses as it represents a typical APAS (Sauerwein, Antensteiner, et al., 2023). To ensure consistent results, according to Sauerwein, Antensteiner, et al. (2023) a standardized blended learning approach was implemented across all courses where students were provided with a weekly exercise sheet through ArTEMiS to support their learning outside the classroom. These sheets comprised (*i*) programming tasks students worked on individually, accompanied by (*ii*) predefined test cases to give them formative feedback within ArTEMiS (Sauerwein, Antensteiner, et al., 2023). Furthermore, following the completion of the exercises, weekly video conferences were conducted for students to discuss their results and receive personalized assistance from instructors. Due to the pandemic, it was not possible to conduct these meetings in person at the university, as stated by Sauerwein, Antensteiner, et al. (2023).

According to Sauerwein, Antensteiner's, et al. (2023) data set, a total of 414 students took part in their survey, with over 75% of them pursuing a Bachelor's degree, around 20% enrolled in a Master's program, and 5% holding a Master's degree. After removing 132 partially completed questionnaires that contained missing values in either the demographics' data or the research model items columns, 282 fully completed records remained for our statistical analysis. Therefore, compared to Sauerwein, Antensteiner's, et al. (2023) study, which is based on a cleaned dataset of 368 records, our multigroup analysis draws on 86 fewer records for the group of students having used the ArTEMiS Environment, potentially leading to slightly different model evaluation results (cf. Section 5.1).

Among the participants, 75 identified as female, 200 as male, and 7 did not indicate their gender. Approximately 45% of the students, amounting to 122, were not familiar with well-known programming languages like C, Java, or Python. Out of the surveyed students, 95 reported knowledge of Python, 68 had heard of Java, 61 were acquainted with C, and 46 were familiar with C++ prior to enrolling in the respective introductory programming course.

4.4. Data Collection Jupyter Notebook Environment

While the original study described in Section 4.3 was conducted at public universities, the replication study was performed at a so-called University of Applied Sciences. These institutions are part of the Austrian tertiary education system and coexist with public universities, yet they have different focuses as they emphasize the teaching of professional practice at the university level and the conduct of applied research. In terms of education, Universities of Applied Sciences are organized more like schools, with tighter integration of cohorts and students attending the same courses per semester. Moreover, completing all semester courses successfully within a given term is mandatory and must be achieved by the subsequent term's end. Consequently, this often results in shorter overall study periods.

The APAS employed in the replication study is a JupyterHub installation, which provides Jupyter Notebooks to multiple students over the web using a server cluster (Salhofer, 2023). To facilitate assignment management, sharing, collection, and automated grading based on notebooks, the system incorporates the extension nbgrader (Jupyter et al., 2019).

A Jupyter Notebook is an interactive web page that consists of cells, which can contain text descriptions or code snippets. It provides a convenient environment for writing and executing code only using a web browser, offering immediate feedback to programmers and encouraging experimentation with the programming language at hand. With built-in features like syntax highlighting and code completion, there is no need to set up a separate integrated development environment or install any programming language locally.

In the context of assignments, code blocks are provided for students to complete. Following these blocks, there are cells that test the code written by students, giving them instant feedback on the correctness of their solutions. Moreover, it is possible

Item	Survey Question
MOT 1	The APAS motivates me to work on the tasks and the feedback.
MOT 2	Working with the APAS increases my MOT to solve tasks.
MOT 3	The APAS increases my MOT when learning the course content.
INQ 1	The information provided by the APAS is relevant to my learning.
INQ 2	The information provided by the APAS is easy to understand.
INQ 3	The information provided by the APAS is very good.
INQ 4	The information provided by the APAS is up to date.
INQ 5	The information provided by the APAS is complete.
INQ 6	The information provided by the APAS is accurate.
SYQ 1	The APAS offers flexibility in terms of learning time and place.
SYQ 2	The APAS offers interactive feedback on the exercises.
SYQ 3	The APAS offers sufficient functions for my learning process.
SYQ 4	In general, the APAS is reliable.
SYQ 5	In general, the response time of the APAS is reasonable.
SEQ 1	The help and support of the APAS is fast.
SEQ 2	The help and support of the APAS is reliable.
SEQ 3	The help and support of the APAS is accessible.

Item	Survey Question
SEQ 4	The help and support of the APAS is practical.
SEQ 5	The help and support provided by the APAS is satisfactory overall.
LEC 1	I like to use the APAS to support my learning process
LEC 2	Working with the APAS is fun.
LEC 3	Working with the APAS is very pleasant
LEC 4	The LEC of the APAS was able to motivate my spontaneous learning.
BLF 1	The APAS gives me more flexibility to learn in this course.
BLF 2	The APAS allows me to complete the exercises at my own pace.
BLF 3	The APAS allows me to prepare more effectively for each lesson.
TTF 1	The services offered by the APAS meet my learning needs.
TTF 2	The services of the APAS are compatible with my learning needs.
TTF 3	The services of the APAS fit my learning needs.
TTF 4	My learning goals and needs are met through the use of the APAS.
SAT 1	Overall, I am satisfied with the APAS.
SAT 2	Overall, I find the APAS enjoyable
SAT 3	Overall, the APAS meets my learning needs.

Table 2. Online questionnaire's survey items (Sauerwein, Antensteiner, et al., 2023).

to include additional tests that are not part of the students' notebook version and are only evaluated during automated grading. Once the grading process is complete, an automatically generated feedback is released to each student, indicating the points earned for each exercise.

Although JupyterHub is a powerful and scalable system, it does not natively support multiple groups within a single course. To address this limitation, several custom extensions have been developed. These extensions facilitate the administration of different programming languages, courses, and student assignments, as well as the mapping of students to specific courses and groups (Salhofer, 2023).

During the end of the summer semester 2023 we replicated Sauerwein, Antensteiner's, et al. (2023) study (cf. Section 4.3) in both a Bachelor's and a Master's program at an Austrian University of Applied Sciences strongly focused on computer science. The 33 survey items defined by Sauerwein, Antensteiner, et al. (2023) were reused (cf. Table 2), however, certain questions regarding the demographics required slight rephrasing to better align with the institution's context. Aligned with (Sauerwein, Antensteiner, et al., 2023), our empirical research centered on introductory programming courses that are suitable for the use of an APAS. Additionally, these courses adopted a learning environment similar to the one described in Section 4.3, which entailed weekly assignments that were automatically graded and discussed in class. Consequently, our research differs notably from Sauerwein, Antensteiner's, et al. (2023) in that (*i*) students had the opportunity to present their solutions in class and engage in face-to-face discussions in subsequent sessions, and (*ii*) Jupyter Notebooks provided them with formative feedback.

A total of 88 students – about 70% were seeking a Bachelor's degree, more than 25% were enrolled in a Master's program and 5% hold a Master's degree – participated in our survey. After removing a total of 16 partially completed questionnaires that contained missing values in either the demographics' data or the research model items columns (cf. Section 4.3), 72 records remained for statistical analysis.

Of the overall 20 female, 50 male students, and 2 with no indication of gender who participated in our survey, about 40%, i.e., in total 29, were not familiar with any well-known (general-purpose) programming language such as C++, Java or Python. Taken together, 21 of the students surveyed stated to know Python, less than one third, i.e., overall 21, have heard of Java and 18 were familiar with C++.

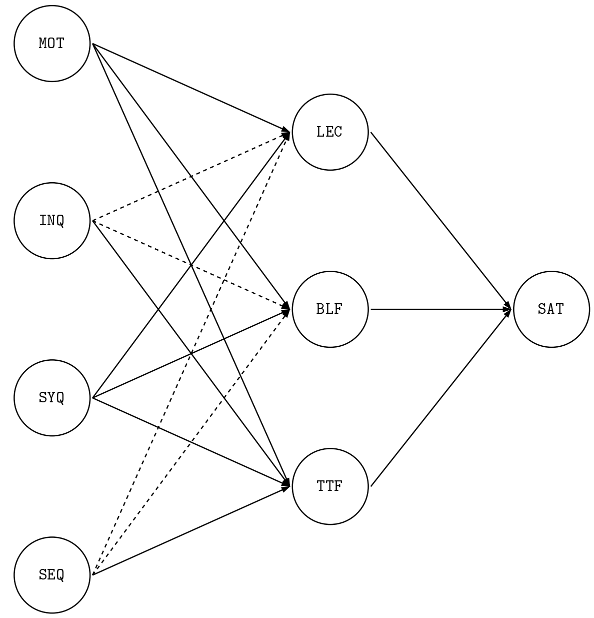

Figure 1. Full APASs' success model.
Assessment of the structural model's path coefficients' significance draws on the full set of data, i.e., Sauerwein, Antensteiner's, et al. (2023) data and the one from our replication study. Solid arrows denote significant paths, while dotted arrows indicate non-significant paths.

5. Data Analysis and Results

To perform a multigroup analysis (cf. Section 5.3) based on Sauerwein, Antensteiner, et al. (2023) theoretically established model of APASs' success and hypothesized relationships (cf. Section 4.1 (*i*)), in a first step, we estimated and evaluated the three corresponding structural equation models (Hair Jr, Hult, Ringle, & Sarstedt, 2021) (cf. Section 5.1).

In a second step, we employed the measurement invariance of composite models (MICOM) procedure (Henseler et al., 2016) to confirm partial measurement invariance (Hair Jr et al., 2017) (cf. Section 5.2). As a final step, we draw on the non-parametric partial least squares multigroup analysis (PLS-MGA) (Henseler, 2007) to test our second hypothesis (cf. Section 4.1 (*ii*)) that the path coefficients between the two groups of students having used either the ArTEMiS Environment (cf. section 4.3) or the Jupyter Notebook Environment (cf. section 4.4) are not significantly different (cf. Section 5.3).

5.1. Models Evaluation

Structural equation modeling (SEM) is a valuable method for evaluating complex theoretical relationships, in particular between multiple latent constructs, especially when undertaking investigations, such as our study, in the social sciences (Hair & Alamer, 2022). While two major SEM methods have been proposed – covariance-based SEM and PLS-SEM – PLS-SEM is considered the preferred SEM technique when making predictions and explanations or, as in our case, testing and validating theories (Shmueli et al., 2016). Consequently, we employed the statistical computing language R version *4.3.* and the `SEMinR` package (Ray et al., 2021) version *2.3.* to test our first hypotheses (cf. Section 4.1 (*i*)) using a PLS-SEM analysis (Hair Jr, Hult, Ringle, Sarstedt, et al., 2021; Ray et al., 2021).

Beginning with the reliability and validity assessment of our reflective measurements by following Hair Jr, Hult, Ringle, Sarstedt, et al. (2021), we (*i*) determined indicator reliability using a threshold value of .708 (Hair Jr, Hult, Ringle, & Sarstedt, 2021) for the factor loadings by eliminating *INQ* 4, *SYQ* 1, as well as *SYQ* 5 (Hair Jr, Hult, Ringle, Sarstedt, et al., 2021). Moreover, we (*ii*) examined internal consistency reliability as the composite reliability (ρ_c), Cronbach's alpha (ρ_T) as well as the reliability coefficient (ρ_A) are all above .70 and less than .95 (Hair Jr, Hult, Ringle, Sarstedt, et al., 2021). According to Hair and Alamer (2022), we (*iii*) gauged convergent validity given that the average variance extracted (AVE) values are all above the cut-off value of .50. Furthermore, as the heterotrait-monotrait (HTMT) ratios are below the liberal cut-off of .90 (Hair & Alamer, 2022), (*iv*) discriminant validity is not a concern (Hair Jr & Sarstedt, 2019).

As recommended by Hair and Alamer (2022), to examine the structural component of the APASs'

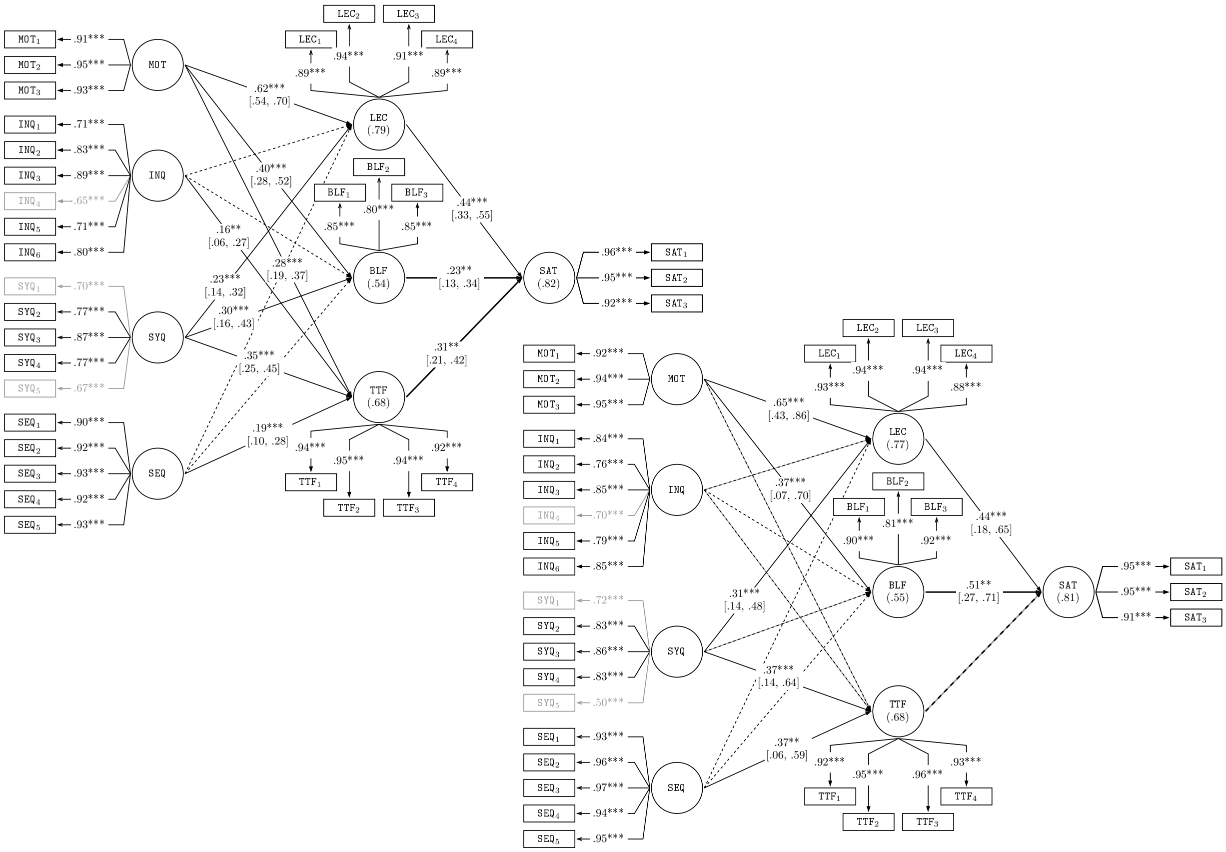

Figure 2. ArTEMiS Environment (Sauerwein, Antensteiner, et al., 2023) success model (left) and Jupyter Notebook Environment (right) success model.

Coefficients of non-significant paths, i.e., dotted arrows, are not shown. Items marked in gray refer to eliminated indicators. Values in braces denote the R^2 value of the respective (endogenous) construct, while value pairs in brackets indicate the 95% confidence interval of the corresponding significant path coefficient. Stars refer to the significance level of the respective factor loading or path coefficient, where three stars ("***") denotes $p \in [0, .001]$ and two stars ("**") denotes $p \in (.001, 0.01]$. Path coefficients no longer significant compared to the full APASs' success model (cf. Figure 1) are indicated by gray and black dotted arrows, while significantly different path coefficients in both APAS Environment-specific success models are emphasized.

success model, we (*i*) examined that our models do not show signs of collinearity among their eight constructs (Hair Jr, Hult, Ringle, Sarstedt, et al., 2021), as the highest variance inflation factor (VIF) values are close to the conservative threshold of 3.3 (Hair & Alamer, 2022). Moreover, we (*ii*) evaluated the significance as well as the relevance of the structural model paths by bootstrapping the sampling distributions with 10 000 resamples (Hair Jr, Hult, Ringle, Sarstedt, et al., 2021) to test the structural model's path coefficients for statistical significance, as summarized in Figure 1 and Figure 2. Following Hair and Alamer (2022), we (*iii*) evaluated a good in-sample predictive power (Hair Jr, Hult, Ringle, & Sarstedt, 2021) as the respective R^2 values are both substantial and moderate (Hair Jr, Hult, Ringle, Sarstedt, et al., 2021).

5.2. Measurement Invariance Assessment

Since, according to Hair Jr et al. (2017), an important prerequisite for comparing group-specific parameter estimates for significant differences using PLS-MGA (Henseler, 2007) is to ensure measurement invariance, we used the `csem` package (Rademaker & Schuberth, 2020) version *0.5.* to perform the MICOM procedure (Henseler et al., 2016) to confirm partial measurement invariance (Hair Jr et al., 2017).

Given that the measurement models for both groups of students consist of the identical items and constructs, data processing, and algorithm settings, following Hair Jr et al. (2017), we (*i*) observed that configural invariance is established. Furthermore, we (*ii*) ascertained partial measurement invariance, as the correlation between the composite scores is not significantly different from 1 (Henseler et al., 2016), i.e., all p values are above .05.

By establishing measurement invariance, we can be confident that group differences in model estimates are not due to differences in the content or meaning of the latent variables formed by the two groups of students having used either the ArTEMiS Environment (cf. section 4.3) or the Jupyter Notebook Environment (cf. section 4.4) (Henseler et al., 2016). Consequently, following Hair Jr et al. (2017), we can utilize a multigroup analysis to compare the path coefficients.

5.3. Multigroup Analysis

Building on the bootstrapped results of the two groups of students, we employed Henseler's (2007) non-parametric PLS-MGA using 10 000 bootstrap samples to show that just the path coefficients between *BLF* and *SAT* ($\text{diff}_{BLF \to SAT} \approx .27, 1 - p_{BLF \to SAT} \approx 1 - .98 < \text{CI}_\alpha$) as well as *TTF* and *SAT* ($\text{diff}_{TTF \to SAT} \approx .27, p_{TTF \to SAT} \approx .01 < \text{CI}_\alpha$) are significantly different at $\text{CI}_\alpha = .05$, respectively (cf. Figure 2).

6. Discussion

Our data analysis (cf. Section 5) confirmed the soundness of both the full and the two APAS Environment-specific reflective measurement models as well as the structural models. Accordingly, in the following, we discuss the main results (cf. Section 6.1) of the models' evaluation (cf. Section 5.1) as well as the multigroup analysis (cf. Section 5.3). Furthermore, we critically analyze the results while taking into account their limitations (cf. Section 6.2).

6.1. Key Findings

Consistent with the hypothesis made by Sauerwein, Antensteiner, et al. (2023), *MOT* and *SYQ*, respectively, have an indirect effect on *SAT* through *LEC*, *BLF*, and *TTF* in the full APASs' success model (cf. Figute 1). Accordingly, based on the data from Sauerwein, Antensteiner, et al. (2023) and our replication study, it appears that *INQ* no longer is a critical success factor of APASs in contrast to *MOT* and *SYQ*.

Moreover, the relevance of *INQ* has diminished in both the full APASs' success model (cf. Figure 1) and the two APAS Environment-specific success models (cf. Figure 2), despite being identified as a critical success factor for APASs by Sauerwein, Antensteiner, et al. (2023) (cf. Section 4.3). They argued that the quality of exercise sheets, along with appropriate feedback, is essential for realizing the full potential of APASs. However, it is worth considering that the importance of *INQ* may be contingent on the usage patterns of APASs as recent research has indicated that APASs, such as ArTEMiS, are predominantly utilized as submission systems (Sauerwein, Priller, et al., 2023). Consequently, our empirical study emphasizes that the quality of the APAS's output or content is subordinate to the success factors of APASs. This finding challenges the original assertion regarding the relevance of *INQ* on automatically generated, formative feedback.

Furthermore, as described in Section 5.3, only the path coefficients between (*i*) *BLF* and *SAT* as well as (*iI*) *TTF* and *SAT* differ significantly in both APAS Environment-specific success models (cf. Figure 2). These results may be attributed (*i*) partially to the use of the Jupyter Notebook Environment for collaboration and discussion in the classroom at the university (cf. Section 4.4), as opposed to the ArTEMiS Environment, which was utilized during the COVID-19 pandemic (cf. Section 4.3). Additionally, (*ii*) the ArTEMiS Environment utilized an APAS that was more specifically designed for programming education (Sauerwein, Antensteiner, et al., 2023).

6.2. Limitations

The research at hand is subject to certain threats to validity that need to be acknowledged. Firstly, there may be a selection bias among the participants in the study. All participants, including those from the ArTEMiS Environment and Jupyter Notebook Environment, voluntarily took part in the study as part of their study program. In comparison to the study conducted by Sauerwein, Antensteiner, et al., 2023, the replication study included students from a university of applied science who were enrolled in bachelor or master programs. This broader inclusion of participants and study programs resulted in higher heterogeneity of the sample. However, it is important to note that the selected courses in both studies had a strong emphasis on programming or computer science, leading to a sample that is predominantly focused on computer science students. To address this limitation, future research could consider including participants from other fields of study to broaden the spectrum.

Secondly, the participants in the studies might have influenced the results by exhibiting a more positive attitude towards the used APASs, as they were surveyed during the latter half of the courses and there might be the risk of students who dropped out would no longer be included in the sample. Unfortunately, due to organizational constraints, it was not possible to quantify the number of drop-outs and address this issue.

Thirdly, the objective of this study was to assess the effectiveness of the APAS success model using a different APAS in a face-to-face teaching setting. In the replication study, as explained in Section 4.4, a Jupyter Notebook-based APAS was utilized. Given the extensive range of available APAS options (Keuning et al., 2018), it remains arguable that the model can be further examined in the context of other APAS platforms. The limitation of the learning method employed in the original study could be overcome by implementing a blended learning approach that incorporates both online and in-person elements. Unfortunately, the pandemic prevented the original study from incorporating such a methodology.

Lastly, as we reused the success model and survey instruments developed by Sauerwein, Antensteiner, et al. (2023), our study primarily focused on a substantial number of constructs at a more abstract level. Therefore, it is important for future research to shift their attention towards these specific constructs and conduct detailed investigations to gain deeper insights into their dynamics and impacts.

7. Conclusion & Outlook

The goal of this paper was to reevaluate a proposed success model for APASs (Sauerwein, Antensteiner, et al., 2023) in a distinct learning environment employing a different APAS, as Sauerwein, Antensteiner's, et al. (2023) hypothesized success model was originally based on a single system. To accomplish this, we conducted a replication study at a University of Applied Sciences, utilizing a Jupyter Notebook based APAS in a face-to-face teaching setting. In doing so, we addressed the limitations highlighted in Sauerwein, Antensteiner's, et al. (2023) study regarding the used APAS and teaching method. By merging the study results, we have contributed to an enhanced understanding of the success factors associated with APASs across heterogeneous learning environments. Our investigation revealed that the majority of the hypotheses in the proposed success model remain valid, with the exception of the impact of information quality on blended learning flexibility and learning climate, which should be addressed in more detail by follow-up research. Furthermore, it is crucial not only to rely on student perceptions, but also to validate the potential positive impact of using APASs through learning analytics that can assess factors such as better grades and lower dropout rates in future research.

References

Barra, E., Lopez-Pernas, S., Alonso, A., Sanchez-Rada, J. F., Gordillo, A., & Quemada, J. (2020). Automated assessment in programming courses: A case study during the covid-19 era. *Sustainability*, *12*(18), 7451.

DeLone, W. H., & McLean, E. R. (2003). The delone and mclean model of information systems success: A ten-year update. *Journal of management information systems*, *19*(4), 9–30.

English, J., & English, T. (2015). Experiences of using automated assessment in computer science courses. *Journal of Information Technology Education: Innovations in Practice*, *14*, 237–254.

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. *International Conference on Engineering Education–ICEE*, *7*.

Gordillo, A. (2019). Effect of an instructor-centered tool for automatic assessment of programming assignments on students' perceptions and performance. *Sustainability*, *11*(20), 5568.

Hair, J., & Alamer, A. (2022). Partial least squares structural equation modeling (pls-sem) in second language and education research: Guidelines using an applied example. *Research Methods in Applied Linguistics*, *1*(3), 100027.

Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2021). *A primer on partial least squares structural equation modeling (pls-sem)*. Sage publications.

Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Danks, N. P., & Ray, S. (2021). Partial least squares structural equation modeling (pls-sem) using r: A workbook.

Hair Jr, J. F., & Sarstedt, M. (2019). Factors versus composites: Guidelines for choosing the right structural equation modeling method. *Project Management Journal*, *50*(6), 619–624.

Hair Jr, J. F., Sarstedt, M., Ringle, C. M., & Gudergan, S. P. (2017). *Advanced issues in partial least squares structural equation modeling*. SAGE Publications Sage CA: Los Angeles, CA.

Henseler, J. (2007). A new and simple approach to multi-group analysis in partial least squares path modeling. *Proceedings of the 5th International Symposium on PLS and Related Methods*, 104–107.

Henseler, J., Ringle, C. M., & Sarstedt, M. (2016). Testing measurement invariance of composites using partial least squares. *International marketing review*.

Jupyter, P., Blank, D., Bourgin, D., Brown, A., Bussonnier, M., Frederic, J., Granger, B., Griffiths, T., Hamrick, J., Kelley, K., Pacer, M., Page, L., Perez, F., Ragan-Kelley, B., Suchow, J., & Willing, C. (2019). Nbgrader: A tool for creating and grading assignments in the jupyter notebook. *Journal of Open Source Education*, *2*, 32.

Keuning, H., Jeuring, J., & Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, *19*(1), 1–43.

Krusche, S., & Seitz, A. (2018). Artemis: An automatic assessment management system for interactive learning. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 284–289.

Luxton-Reilly, A., Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M. J., Sheard, J., & Szabo, C. (2018). Introductory programming: A systematic literature review. *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 55–106.

Mekterovic, I., & Brkic, L. (2017). Setting up automated programming assessment system for higher education database course. *International Journal of Education and Learning Systems*, *2*.

Mekterović, I., Brkić, L., Milašinović, B., & Baranović, M. (2020). Building a comprehensive automated programming assessment system. *IEEE Access*, *8*, 81154–81172.

Pettit, R., Homer, J., Gee, R., Mengel, S., & Starbuck, A. (2015). An empirical study of iterative improvement in programming assignments. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 410–415.

Pettit, R. S., Homer, J. D., McMurry, K. M., Simone, N., & Mengel, S. A. (2015). Are automated assessment tools helpful in programming courses? *2015 ASEE Annual Conference & Exposition*, 26–230.

Queiros, R., Pinto, M., & Terroso, T. (2020). Computer programming education in portuguese universities. *First International Computer Programming Education Conference*.

Rademaker, M. E., & Schuberth, F. (2020). *Csem: Composite-based structural equation modeling* [Package version: 0.5.0.9000]. https://m-e-rademaker.github.io/cSEM/

Ray, S., Danks, N., & Calero Valdez, A. (2021). Seminr: Domain-specific language for building, estimating, and visualizing structural equation models in r. *Estimating, and Visualizing Structural Equation Models in R (August 6, 2021)*.

Restrepo-Calle, F., Ramirez Echeverry, J. J., & Gonzalez, F. A. (2019). Continuous assessment in a computer programming course supported by a software tool. *Computer Applications in Engineering Education*, *27*(1), 80–89.

Salhofer, P. (2023). An integrated learning and assessment environment for stem courses based on jupyterhub. *INTED2023 Proceedings*, 733–740.

Sauerwein, C., Antensteiner, T., Oppl, S., Groher, I., Meschtscherjakov, A., Zech, P., & Breu, R. (2023). Towards a success model for automated programming assessment systems used as a formative assessment tool. *Proceedings of the 28th annual ACM conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 271–277.

Sauerwein, C., Priller, S., Dobiasch, M., Oppl, S., Felderer, M., & Breu, R. (2023). Lecturers' and students' experiences with an automated programming assessment system.

Shmueli, G., Ray, S., Estrada, J. M. V., & Chatla, S. B. (2016). The elephant in the room: Predictive performance of pls models. *Journal of business Research*, *69*(10), 4552–4564.

Souza, D. M., Felizardo, K. R., & Barbosa, E. F. (2016). A systematic literature review of assessment tools for programming assignments. *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, 147–156.

Ullah, Z., Lajis, A., Jamjoom, M., Altalhi, A., Al-Ghamdi, A., & Saleem, F. (2018). The effect of automatic assessment on novice programming: Strengths and limitations of existing systems. *Computer Applications in Engineering Education*, *26*(6), 2328–2341.

Zhang, Y. G., & Dang, M. Y. (2020). Understanding essential factors in influencing technology-supported learning: A model toward blended learning success. *Journal of Information Technology Education. Research*, *19*, 489.