Educ Inf Technol
https://doi.org/10.1007/s10639-017-9658-2

Exploring computer science students' continuance intentions to use Kattis

Ram B. Basnet[1] · **Tenzin Doleck**[2] · **David John Lemay**[2] · **Paul Bazelais**[2]

Received: 30 July 2017 / Accepted: 10 October 2017
© Springer Science+Business Media, LLC 2017

Abstract Teaching programming concepts to enhance students' problem solving and computational thinking skills is a challenging task, especially when students enter college with little to no preparation, or they lack the interest or capacity for programming. Online platforms that serve as automated practice and assessment systems have been offered as potential tools for supporting programming skills development, providing feedback, and motivating students. The present article discusses the use of an online automated practice and assessment system called Kattis for homework assignments and final project in three computer science courses. The goal of the present study was to ascertain students' continuance intentions to use Kattis. We attempt to address this by using partial least squares on data from a survey of 50 students. The findings of the present study suggest that continuance intentions to use Kattis is driven by students' level of satisfaction with the system, the degree of students' confirmation of expectations, and the perceived usefulness of the system.

Keywords Kattis · Problem solving · Automated assessment · Feedback · Computational thinking · Computer programming · Continuance intentions

✉ Tenzin Doleck
tenzin.doleck@mail.mcgill.ca

Ram B. Basnet
rbasnet@coloradomesa.edu

David John Lemay
david.lemay@mail.mcgill.ca

Paul Bazelais
paul.bazelais@mail.mcgill.ca

[1] Colorado Mesa University, 1100 North Ave., Grand Junction, CO 81501, USA

[2] McGill University, 3700 McTavish St., Montreal, QC H3A 1Y2, Canada

Published online: 17 October 2017

1 Introduction

One of the big challenges in computer science education concerns engaging students who have little to no preparation or have difficulty completing the necessary work to succeed (Stone and Madigan 2008). Programming is a demanding endeavor, and many novice students face several challenges and difficulties in learning to program (Robins et al. 2003). Several researchers have suggested that traditional forms of assessment such as quizzes and exams are not effective for programming courses as they don’t provide sufficient feedback in a course where the primary objectives are learning programming concepts and practicing computational thinking by solving real-world problems and ideally requires support through the whole programming cycle (Enström et al. 2011; Garcia-Mateos and Fernandez-Aleman 2009).

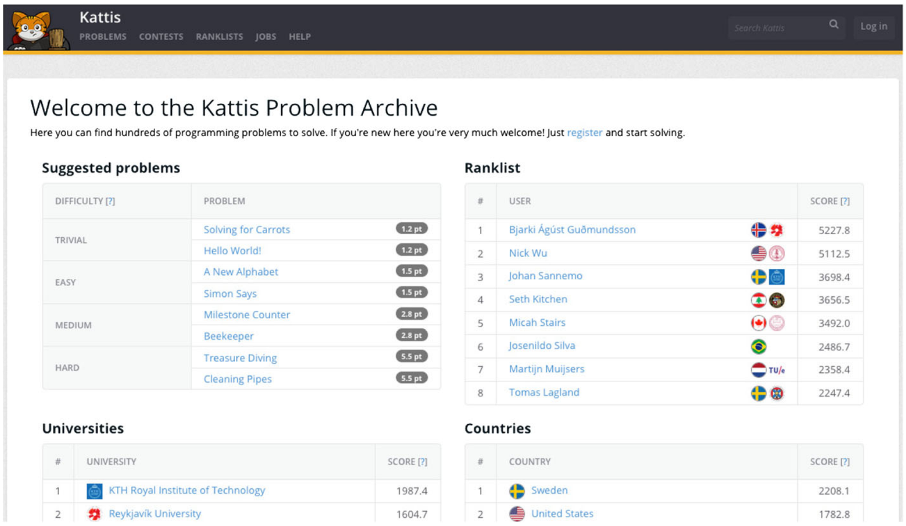
Formative feedback enables students to make informed decisions about their learning (Shute 2008). The literature on formative assessment suggests that, to maximize learning, students need to be provided with options to revise their study approach and take control of their learning (Nicol and Macfarlane-Dick 2006). Shute (2008) reported that effective formative assessment should be non-evaluative, supportive, timely, and specific. Thus, feedback is tailored to be maximally effective and applicable to the student’s practice. The notion of formative assessment is also related to the notion of deliberate practice (Ericsson et al. 1993) that holds that expert performance is derived from practicing skills at the limit of one’s ability, to strengthen and develop skills by maintaining such deliberate practice for extended intervals. Deliberate practice provides the necessary kind of formative data to correct understandings and improve technique. Automated assessments systems (Enström et al. 2011) have been devised to provide the kind of environment where students can practice programming at their level of mastery, and offer the kind of formative feedback to fix holes in their knowledge and practice those concepts and skills that they need to develop to attain mastery. Automated assessment systems offer individualized formative feedback that is scalable and responsive to the individual learner’s level of mastery. Automated assessment systems can vary in their level of adaptability, from offering a range of problem sets, as with Kattis, to providing tailored feedback or adaptive lesson paths based on learner performance using statistical modelling techniques (Piech et al. 2015).

Programming instruction requires authentic designs that incorporate interesting and relevant assignments and emulate real-world problems that are both meaningful and challenging, to motivate students to study (Ala-Mutka 2005; Layman et al. 2007). To succeed in computer science programs, students need to be intrinsically motivated to continuously engage in and practice coding (Bergin and Reilly 2005; Fernandez Aleman 2011). To meet such challenges, automated assessment systems have been offered as viable solutions to improve instruction and learning (Ala-Mutka 2005; Blumenstein et al. 2008; Enström et al. 2011; Wang et al. 2011). Automated assessment platforms such as Kattis (Open Kattis 2017) provide important contexts for authentic coding exposure and are increasingly becoming popular (Enström et al. 2011). Such platforms provide many opportunities for revamping assessment and feedback, and have attracted attention from computer science instructors, researchers, and practitioners (Ala-Mutka 2005; Fernandez Aleman 2011). According to Ala-Mutka (2005), the advantages of automated assessment platforms include “speed, availability, consistency and objectivity of assessment” (p. 83). The objective of this research is to ascertain whether students are self-motivated to challenge themselves and go above and beyond the assigned assignments and projects to continuously practice programming using the Kattis platform.

2 Kattis

Kattis (Fig. 1) is an online service developed and used since 2005 to automate student assessment and grading in computer programming courses at KTH – Royal Institute of Technology, Sweden (Enström et al. 2011). An example of a problem posted on Kattis is illustrated in Fig. 2. Kattis provides thousands of solutions to pick from, developed by professionals and educators around the world. According to Enström et al. (2011), "automated assessment systems clearly assist in reducing the teacher's workload by removing the tedious work for manually verifying correctness" (p. T3 J-1). Started as an academic project at KTH, the service is now used in computer programming courses in universities across the globe and in competitions too. For example the ACM International Collegiate Programming Contest (ICPC) uses Kattis (ICPC Kattis 2017). Kattis has found use beyond academia, companies use it judge and recruit talented programmers. They use the service to test applicants' programming competence and problem solving skills in an automated fashion before inviting them to an in-person interview (Kattis 2017).

Kattis was first used in two courses at KTH Royal Institute of Technology in 2005 and 2006. It has gone through several iterations based on students' feedback and instructors' experiences (Enström et al. 2011). Because Kattis' online interface publishes not only the CPU time a submitted solution takes but also the top 10 lists of fastest solutions in various programming languages, Enström et al. (2011) have noted that students are self-motivated to, perhaps challenge themselves, and submit multiple accepted submissions to the same problems.



Welcome to the Kattis Problem Archive

Here you can find hundreds of programming problems to solve. If you're new here you're very much welcome! Just register and start solving.

Suggested problems

DIFFICULTY [?]	PROBLEM	
TRIVIAL	Solving for Carrots	1.2 pt
	Hello World!	1.2 pt
EASY	A New Alphabet	1.5 pt
	Simon Says	1.5 pt
MEDIUM	Milestone Counter	2.8 pt
	Beekeeper	2.8 pt
HARD	Treasure Diving	5.5 pt
	Cleaning Pipes	5.5 pt

Ranklist

#	USER	SCORE [?]
1	Bjarki Ágúst Guðmundsson	5227.8
2	Nick Wu	5112.5
3	Johan Sannemo	3698.4
4	Seth Kitchen	3656.5
5	Micah Stairs	3492.0
6	Josenildo Silva	2486.7
7	Martijn Muijsers	2358.4
8	Tomas Lagland	2247.4

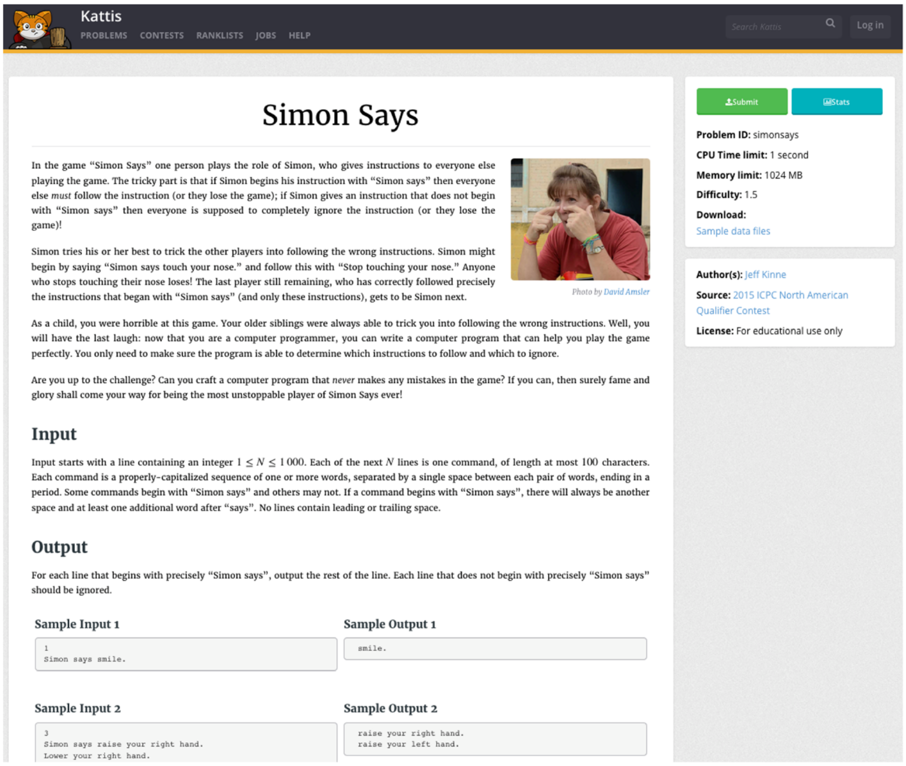
Universities

#	UNIVERSITY	SCORE [?]
1	KTH Royal Institute of Technology	1987.4
2	Reykjavík University	1604.7

Countries

#	COUNTRY	SCORE [?]
1	Sweden	2208.1
2	United States	1782.8

Fig. 1 Kattis Interface. Reprinted from Kattis Problem Archive, Retrieved July 9, 2017, from https://open.kattis.com. Copyright 2017 by Scrool AB. Reprinted with permission



Simon Says

In the game “Simon Says” one person plays the role of Simon, who gives instructions to everyone else playing the game. The tricky part is that if Simon begins his instruction with “Simon says” then everyone else *must* follow the instruction (or they lose the game); if Simon gives an instruction that does not begin with “Simon says” then everyone is supposed to completely ignore the instruction (or they lose the game)!

Simon tries his or her best to trick the other players into following the wrong instructions. Simon might begin by saying “Simon says touch your nose.” and follow this with “Stop touching your nose.” Anyone who stops touching their nose loses! The last player still remaining, who has correctly followed precisely the instructions that began with “Simon says” (and only these instructions), gets to be Simon next.

As a child, you were horrible at this game. Your older siblings were always able to trick you into following the wrong instructions. Well, you will have the last laugh: now that you are a computer programmer, you can write a computer program that can help you play the game perfectly. You only need to make sure the program is able to determine which instructions to follow and which to ignore.

Are you up to the challenge? Can you craft a computer program that *never* makes any mistakes in the game? If you can, then surely fame and glory shall come your way for being the most unstoppable player of Simon Says ever!

Input

Input starts with a line containing an integer $1 \leq N \leq 1\,000$. Each of the next N lines is one command, of length at most 100 characters. Each command is a properly-capitalized sequence of one or more words, separated by a single space between each pair of words, ending in a period. Some commands begin with “Simon says” and others may not. If a command begins with “Simon says”, there will always be another space and at least one additional word after “says”. No lines contain leading or trailing space.

Output

For each line that begins with precisely “Simon says”, output the rest of the line. Each line that does not begin with precisely “Simon says” should be ignored.

Sample Input 1

```
1
Simon says smile.
```

Sample Output 1

```
smile.
```

Sample Input 2

```
3
Simon says raise your right hand.
Lower your right hand.
```

Sample Output 2

```
raise your right hand.
raise your left hand.
```

Problem ID: simonsays
CPU Time limit: 1 second
Memory limit: 1024 MB
Difficulty: 1.5
Download: Sample data files

Author(s): Jeff Kinne
Source: 2015 ICPC North American Qualifier Contest
License: For educational use only

Fig. 2 Example of a Problem available on Kattis. Reprinted from Kattis Problem Archive, Retrieved July 9, 2017, from https://open.kattis.com/problems/simonsays. Copyright 2017 by Scrool AB. Reprinted with permission

Students need to be continually motivated to engage in and practice coding to become competent and develop the necessary skills for a career in programming. It is important to understand the factors that motivate students to continue using an environment such as Kattis because they are a determinant of implementation success. The purpose of this study was to investigate the continuance intentions of computer science students to use Kattis grounded in the Expectation-Confirmation theory of information systems continuance.

3 Continuance intentions

With the growing push to introduce and integrate technology in education, the need to understand why learners accept technology has become an important exercise (Doleck et al. 2017a; Bazelais et al. 2017). Indeed, Jasperson et al. (2005) highlight that technology adoption literature has largely focused on “individuals' preadoption activities, the adoption decision, and initial use behaviors” (p. 527)—reflected in the greater number of studies related to acceptance behaviors in the educational technology literature. While the acceptance of technology is considered critical for the success of any technology implementation (Doleck et al. 2017b; Lemay et al. 2017; Legris et al.

2003; Venkatesh and Davis 2000), the importance of continued use of the technology has also been emphasized (Bhattacherjee 2001; Hong et al. 2006; Jasperson et al. 2005) given that users at times discontinue use after initially accepting the technology (Bhattacherjee 2001; Thong et al. 2006). Indeed, Bhattacherjee (2001) submits that the "long-term viability of an IS and its eventual success depend on its continued use rather than first-time use" (p. 351–352). As such, investigations geared toward understanding continuance intentions have received growing attention in the educational technology literature (Chiu et al. 2007; Lin 2011; Terzis et al. 2013; Wu et al. 2007). The key objective of the present study then is to describe the relationships between the factors that affect continuance intentions to use Kattis. This is an important exercise, given that teachers sometimes impose the use of certain learning systems in the classroom. The introduction and use of a new learning system can engender a wide range of responses from students. Thus, a more nuanced understanding of students' perceptions of the system post-adoption is needed, one that captures whether students accept and continue to use the system. We rely on the widely used theoretical framework of the Expectation-Confirmation model (Bhattacherjee 2001) to examine the factors that affect the usage continuance intentions of Kattis.

4 Theoretical framework

Bhattacherjee (2001) notes that the stream of work that considers "pre-acceptance variables to explain both acceptance and continuance decision" (p. 352) typically fails to "explain why some users discontinue IS use after accepting it initially" (p. 352). To address this gap in the literature, Bhattacherjee (2001) introduced the expectation-confirmation theory of information systems continuance and posited that, post-acceptance, user's intentions to continue using a system or technology (the dependent variable) are driven by three antecedent constructs: user's satisfaction; the degree of users' expectation confirmation; and, the perceived usefulness of the system or technology. To yield insights into the continued use of Kattis, consistent with the expectation-confirmation theory (Fig. 3), the present study proposed hypotheses that are formalized as follows:

- H1: satisfaction (SAT) is positively related to usage continuance intentions (CIN)
- H2: confirmation (CON) is positively related to satisfaction (SAT)

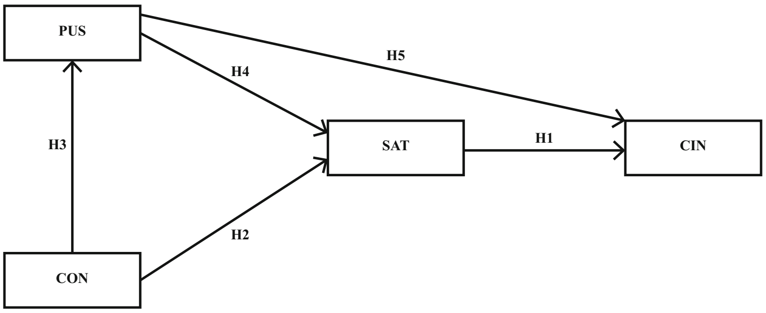


Fig. 3 Expectation-confirmation theory

- H3: confirmation (CON) is positively related to perceived usefulness (PUS)
- H4: perceived usefulness (PUS) is positively related to satisfaction (SAT)
- H5: perceived usefulness (PUS) is positively related to continuance intention (CIN)

5 Method

5.1 Context of this study

This study was conducted in the context of three computer science courses (CSCI 112: Data Structures, CSCI 310: Advanced Programming: Python, CSCI 420: Cybersecurity) in spring 2017. These courses ranged from freshman to senior level courses. The open problem archive service (http://open.kattis.com) of Kattis was used in all the courses. Students were asked to register using their university email id, and were given the choice to use the service in anonymous mode or the default public mode. In the anonymous mode, users can hide their information and ratings from all other users, and their score is not used to calculate the overall rank of the university, subdivision, or the country they belong to. Problems with various difficultly levels (typically ranging from 1.2 to 9.5) were assigned to the students in the three courses as detailed below. Since the difficulty level keeps changing (can go up or down), students were asked to record the difficulty level of the problem at the beginning when they started to solve a problem of their choice. Students could submit the solutions of a problem as many times as they wished, and were encouraged to test their solutions with corner test cases before submitting to the online judge. Only the accepted solutions were graded and no partial credits were given for partially correct solutions (passes in some test cases but fails in at least 1).

In CSCI 112, Kattis was used for the final project. Students were asked to pick and solve problems of varying difficulty level to aim for each letter grade. For example, to receive 100% (A), students needed to solve 3 problems (1 with difficulty level 2 or above; 1 with difficulty level 1.8 or above; and, 1 with difficulty level 1.7 or above); to receive 80% (B), students needed to solve 3 problems (1 with difficulty level 1.7 or above; 1 with 1.6 or above; and, 1 with 1.5 or above), and so on. 72% of the students received an A; 9% received a B and the rest received an F.

The experience of automated assessment systems can be frustrating for users, especially if they're not familiar with online judging system and are beginner programmers. Difficult problems can appear deceivingly simple. Slow and simple recursive and brute-force algorithms can simply fail with the hidden test cases. And the time constraint imposed by Kattis can provoke anxious reactions. All of which can engender a frustrating experience, especially when Kattis intentionally doesn't provide specific feedback. To address this scenario, students were explicitly advised to initially target the C grade to make themselves familiar with the system and, once they gain momentum, to continue solving increasingly difficult problems to target an A grade. We did not seek to confirm if the students took advantage of the strategy. However, many students solved more problems, and some with higher difficulty levels, than the minimum requirements of the project.

In CS310, Kattis was used for the first half of the semester, each student was asked to solve six problems in total with difficulty level from 1.3 and above with the constraint that no two problems could be of the same difficulty level. All students except one (who did not submit the project) met most of the requirements of the project and received an A grade for the Kattis problems. Students continued to solve problems with varying difficulty levels and some were involved in friendly competition among themselves vying for the better rank. Ultimately, the most top ranked students were in this course.

In CSCI420, where C programming language is briefly covered, students were asked to pick and solve one problem with difficulty level 1.5 or above in C programming language as a homework assignment. Although most students in the course were excited about Kattis, one student solved many problems using C# and remained in the top-five institutional ranking for several months. The student was majoring in computer information systems where coding is not even the primary focus. Every student received an A grade in the Kattis assignment.

5.2 Participant profile

Participants in the study were 50 students from a southwestern university who volunteered to participate. The students were drawn from three computer science courses in spring 2017. Table 1 summarizes the data regarding the courses and the participants.

5.3 Instrument

The survey instrument for specifying the factors affecting Kattis continuance intentions was developed using items from the literature (Bhattacherjee 2001; Davis 1989; Lee 2010; Limayem et al. 2007). There were five items for perceived usefulness (e.g., “Using Kattis enables me to accomplish my learning tasks more quickly”), three items for confirmation (e.g., “My experience with using Kattis was better than what I expected”), three items for satisfaction (e.g., “I am satisfied with the performance of Kattis”), and three items for continuance intentions (e.g., “I intend to continue to using Kattis rather than use any alternative”). All items were scored on a 7-point Likert-type rating scale (1 = *strongly disagree* to 7 = *strongly agree*). Participants were also asked to provide general comments about Kattis.

Table 1 Participant summary

Course	# of Students enrolled	# of Surveys	% Male	% Female	% CS Major	Other Major	Seniors & Graduates	Freshman & Junior
CSCI 112	24	21	83%	17%	78%	12%	8%	92%
CSCI 310	29	24	87%	13%	90%	10%	30%	70%
CSCI 420	6	5	64%	36%	85%	15%	100%	0%

6 Analysis and results

We used partial least squares structural equation modeling (PLS-SEM; Hair et al. 2011; Henseler et al. 2016) to test the research hypotheses. All analyses were carried out using the WarpPLS tool (Kock 2015a; Kock 2015b). We followed the standard two-step modeling process: measurement model and structural model (Hair et al. 2011; Henseler et al. 2016; Kock 2015b). The psychometric properties were examined using guidelines from the literature on PLS (Hair et al. 2011; Henseler et al. 2016; Kock 2015b).

6.1 Measurement model

Using the suggested recommended criteria for model fit determination and quality indices, there was acceptable fit of the data (Table 2) to the hypothesized model (Kock 2015b).

The factor loadings all exceeded 0.70 (Chin 1998), presenting a good indicator of the instrument's reliability (Table 3). Table 4 illustrates that the composite reliability (Kock 2015b) coefficients of the different measures all exceeded the threshold value of 0.70 and the Cronbach's alpha coefficients (Kock 2015b) of the different measures all exceeded the threshold value of 0.70. Thus, establishing the reliability of the indicators. Convergent validity was assessed through the average variance extracted (AVE) test on the variables. The values in Table 4 support convergent validity as all AVEs exceeded the recommended threshold value 0.50 (Henseler et al. 2016).

Discriminant validity was assessed using the Fornell-Larcker criterion (Fornell and Larcker 1981). In Table 5, all the diagonal values (square roots of AVEs) are greater than the off-diagonal numbers in the corresponding rows and columns, and demonstrate discriminant validity.

The acceptability of the psychometric properties of the measurement model established, we turn our attention to the structural model.

Table 2 Model fit statistics and quality indices

Measure	Values	Recommended criterion
Average path coefficient (APC)	0.489, $P < 0.001$	Acceptable if $P < 0.05$
Average R-squared (ARS)	0.626, P < 0.001	Acceptable if $P < 0.05$
Average adjusted R-squared (AARS)	0.613, P < 0.001	Acceptable if $P < 0.05$
Average block VIF (AVIF)	3.060	Acceptable if <= 5
Average full collinearity VIF (AFVIF)	3.193	Acceptable if <= 5
Tenenhaus GoF (GoF)	0.714	small > = 0.1, medium > = 0.25, large > = 0.36
Sympson's paradox ratio (SPR)	1.000	acceptable if > = 0.7
R-squared contribution ratio (RSCR)	1.000	acceptable if > = 0.9
Statistical suppression ratio (SSR)	1.000	acceptable if > = 0.7
Nonlinear bivariate causality direction ratio (NLBCDR)	1.000	acceptable if > = 0.7

Table 3 Loadings of measurement items

	PUS	CON	SAT	CIN	*P* value
PUS1	**0.902**	0.166	0.086	−0.065	<0.001
PUS2	**0.954**	−0.027	0.017	−0.157	<0.001
PUS3	**0.832**	−0.232	−0.119	−0.085	<0.001
PUS4	**0.911**	0.014	−0.057	0.162	<0.001
PUS5	**0.915**	0.061	0.063	0.144	<0.001
CON1	0.562	**0.855**	−0.140	−0.202	<0.001
CON2	−0.157	**0.886**	−0.120	−0.040	<0.001
CON3	−0.403	**0.848**	0.267	0.096	<0.001
SAT1	−0.193	−0.072	**0.870**	−0.239	<0.001
SAT2	0.031	−0.037	**0.934**	−0.105	<0.001
SAT3	0.159	0.112	**0.869**	0.352	<0.001
CIN1	−0.031	0.058	−0.068	**0.974**	<0.001
CIN2	−0.026	−0.031	−0.064	**0.935**	<0.001
CIN3	0.059	−0.030	0.135	**0.934**	<0.001

Bold values are loadings

6.2 Structural model

The full collinearity variance inflation factors (VIFs) were assessed for common method bias test and to detect multicollinearity. Since all VIFs were below the suggested threshold of 5, we concluded that there was no multicollinearity and no common method bias (Kock 2015b). Additionally, the predictive relevance associated with each endogenous variable in the model was examined. All Q^2 coefficient values were found to be greater than zero, demonstrating an acceptable level of predictive relevance (Kock 2015b). Figure 4 illustrates the results of the path estimation results using WarpPLS.

According to Hair et al. (2011), R^2 (coefficient of determination) values of 0.75, 0.50, and 0.25 are considered substantial, moderate, and weak, respectively. Given the R^2 of CIN was 0.60 (moderate), the research model explained 60% of the variance in CIN. More specifically, with an R^2 of 0.60 for CIN, the two latent variables (PUS and SAT) explain 60% of the variance in USE. With an R^2 of 0.76 (substantial) for SAT, the two latent variables (PUS and CON) explain 76% of the variance in SAT. Finally, with an R^2 of 0.51 (moderate) for PUS, the latent variable (CON) explains 51% of the variance in PUS.

Table 4 Measurement scale characteristics

Construct	Composite reliability (CR)	Cronbach's alpha	Average variance extracted (AVE)
PUS	0.957	0.943	0.817
CON	0.898	0.829	0.745
SAT	0.921	0.871	0.795
CIN	0.964	0.944	0.899

Table 5 Discriminant validity check

	PUS	CON	SAT	CIN
PUS	**0.904**	0.651	0.801	0.749
CON	0.651	**0.863**	0.747	0.491
SAT	0.801	0.747	**0.892**	0.728
CIN	0.749	0.491	0.728	**0.948**

Bold values are loadings

The path coefficients (*β*) and path significance (*p*-value) were examined to reveal the relationships between the constructs in the research model. The results of the hypotheses testing including effect sizes (f^2) are presented in Table 6. Values of 0.35, 0.15, and 0.02 are deemed as large, medium, and small, respectively (Cohen 1988). Using the guidelines for assessing effect sizes, most effect sizes in the study were deemed large.

7 Discussion

Drawing on Bhattacherjee's (2001) expectation-confirmation theory of information systems to examine the interplay of the constructs of the research model (Fig. 3), the present study attempted to provide insights into the structural relationships between the constructs (perceived usefulness, satisfaction, and confirmation) and their resulting effect on continuance intentions to use Kattis. The results provide strong empirical support for the proposed model. The analysis yielded significant results in the hypothesized direction for all the hypotheses. Overall, the structural model helped explain 60% (R^2 of CIN was 0.60%) of variance in continuance intentions to use Kattis, thus highlighting the important role that the antecedent variables had on predicting the continuance intentions. First, in support of the two direct antecedent influences (H1 and H5), both perceived usefulness ($\beta = 0.490$, $p < 0.001$) and satisfaction ($\beta = 0.319$, $p = 0.007$) had a positive significant effect on continuance intentions. The antecedent variables, confirmation and perceived usefulness, helped explain 76% (R^2 of SAT was 0.76%) of the variance in satisfaction,

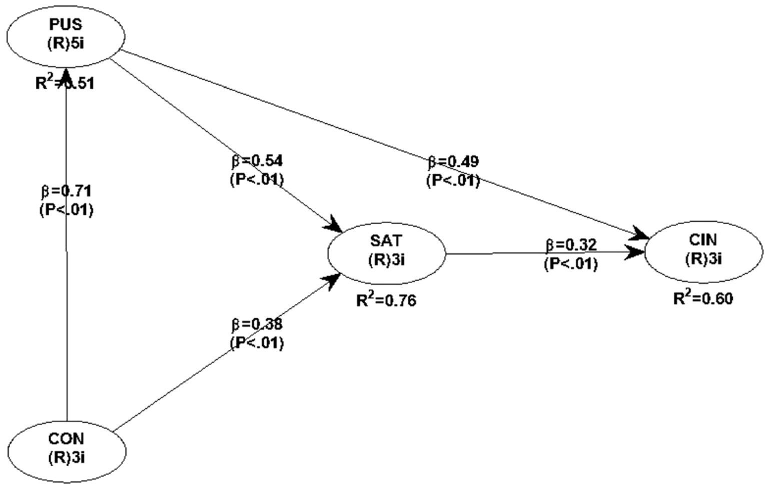


Fig. 4 PLS Results

Table 6 Hypotheses Testing

Hypothesis	Path	Path coefficient (β)	*P value*	Effect size (f^2)	Result
H1	SAT ➔ CIN	0.319	$p = 0.007$	0.233	Supported
H2	CON ➔ SAT	0.380	$p = 0.002$	0.307	Supported
H3	CON ➔ PUS	0.714	$p < 0.001$	0.510	Supported
H4	PUS ➔ SAT	0.540	$p < 0.001$	0.455	Supported
H5	PUS ➔ CIN	0.490	$p < 0.001$	0.371	Supported

thus indicating that confirmation and perceived usefulness were important in satisfaction formation. Second, in support of the two direct antecedent influences (H2 and H4), both confirmation ($\beta = 0.380, p = 0.002$) and perceived usefulness ($\beta = 0.540, p < 0.001$) had a positive significant effect on satisfaction. Third, confirmation had a positive significant effect on perceived usefulness (β = 0.714, p < 0.001) and helped explain 51% of the variance in perceived usefulness; thus supporting H3. In summary, all proposed hypotheses were supported. This study affirms that user's intentions to continue using a system are driven by three antecedent constructs: user's level of satisfaction with the system; the degree of users' confirmation of expectations; and the perceived usefulness of the system. These findings are consistent with much of the research that highlights the influence of the three antecedent constructs on continuance intentions and supports the expectation-confirmation model (Bhattacherjee 2001).

The present study is of practical importance to educators considering the use of open programming and assessment environments like Kattis. The success of adopting any new technology depends on subsequent continued use of the technology. Our findings reveal that the students sampled in the present study were likely to continue to use Kattis and the model provides us an accessible way to understand the reasons why. By revealing the interplay of the antecedent factors to predict continuance intentions, the model can help instructors plan theirs courses to maximally support students in adopting and continuing to use Kattis and to ensure their success in learning programming.

Interestingly, when we examine the relative importance of the direct antecedents to continuance intentions in the research model, we find that perceived usefulness was a stronger predictor of continuance intention compared to satisfaction. This suggests that students are willing to put up with quite a bit of dissatisfaction as they struggle to learn programming. Students appear to value the usefulness of Kattis for learning programming. Despite the hardships, they overwhelmingly report that they will continue to use Kattis and that they see it as an important learning tool.

We find corroborating evidence in the general comments regarding Kattis provided by the students—in addition to reporting on their continuance intentions. Students generally had positive comments regarding the use of Kattis, which accords with the high level of satisfaction and perceptions of usefulness noted in the research model. This finding is echoed in the general positive comments about the usefulness of Kattis.

Some examples are presented below:

"Best part: Employers ask you kattis questions." [P2]

"Nice for HW assignments." [P4]

"The best part of using Kattis is the variety of problems. I think it prepares students for real-world situations." [P7]

"It is really good to try kattis problem as it enhances problem solving skills." [P11]

"I love spending my time on Kattis. It is very helpful to learn better programming." [P13]

"I really enjoyed it and experienced a question at one of my interviews." [P29]

"I think it's great. An in-class competitive aspect would be fun." [P38]

"I really enjoyed using Kattis. I have recommended to other students. Kattis helped me problem solve." [P39]

"The best part about Kattis was using critical thinking to understand and apply the problem to a coding language." [P46]

A few concerns and complaints were also noted, mainly related to the lack of feedback in Kattis:

"The worst part: You don't get to see the test case that is breaking your solution." [P2]

"My only problem is that it doesn't provide helpful feedback." [P22]

"Not much explanation of why solution is wrong." [P23]

"Worst aspect: sometimes doesn't give you complete set of test cases, especially with higher difficulty problems." [P47]

Student comments can provide Kattis developers with helpful guidelines for system improvement and help them to address specific issues with Kattis. Indeed, improving the feedback mechanisms in Kattis ought to help improve perceived usefulness, and in turn improve user satisfaction with the system. Piech et al. (2015) have proposed automated feedback mechanism using program embeddings, that is, representing programs as compositional sequences in a vector space, to provide automated feedback based on semantic similarity of the encoded program. These kinds of probabilistic word vector representations have shown strong results in representing sentence meanings by capturing sentence compositionality, and parsing sentence structures for POS tagging, topic modelling, and sentiment analysis. Such work holds the promise of developing collaborative filtering for automated feedback generation; thereby making system feedback more informative through a scalable process.

The cross-sectional study has several limitations that need to be considered when interpreting the findings. It should be noted that we relied on the original formulation of the expectation-confirmation theory, using the native constructs only. Other salient constructs could also influence the continuance intentions. Consideration of other salient constructs is needed in future studies to better capture the phenomenon. The sample for our study was limited in size. It was drawn from a single college (computer science students), and the courses had male-skewed gender imbalance. Given the cross-sectional nature of the study, a natural concern relates to the generalizability of our findings. Future studies should be conducted with other samples addressing the noted limitations, including longitudinal studies that can verify user continuance intentions and capture changes over time. Jasperson et al. (2005) refer to post-adoption behavior as "myriad feature adoption decisions, feature use behaviors, and feature extension behaviors made by an individual user after an IT application has been installed, made accessible to the user, and applied by the user in accomplishing his/her work activities" (p. 531). Thus, future research needs to examine feature use and extension behaviors as well, to gain better insights into user continuance behaviors.

References

Ala-Mutka, K. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education, 15*(2), 83–102. https://doi.org/10.1080/08993400500150747.

Bazelais, P., Doleck, T., & Lemay, D. J. (2017). Investigating the predictive power of TAM: A Case Study of CEGEP Students' Intentions to Use Online Learning Technologies. *Education & Information Technologies*. Advance online publication. https://doi.org/10.1007/s10639-017-9587-0.

Bergin, S., & Reilly, R. (2005). The influence of motivation and comfort-level on learning to program. In *Proceedings of the 17th workshop on psychology of programming* (pp. 293–304). Brighton: University of Sussex.

Bhattacherjee, A. (2001). Understanding information systems continuance: An expectation-confirmation model. *MIS Quarterly, 25*(3), 351–370. https://doi.org/10.2307/3250921.

Blumenstein, M., Green, S., Fogelman, S., Nguyen, A., & Muthukkumarasamy, V. (2008). Performance analysis of GAME: A generic automated marking environment. *Computers & Education, 50*(4), 1203–1216. https://doi.org/10.1016/j.compedu.2006.11.006.

Chin, W. W. (1998). The partial least squares approach for structural equation modeling. In G. A. Marcoulides (Ed.), *Modern methods for business research* (pp. 295–336). Mahwah, NJ: Erlbaum.

Chiu, C., Sun, S., Sun, P., & Ju, T. (2007). An empirical analysis of the antecedents of web-based learning continuance. *Computers & Education, 49*(4), 1224–1245. https://doi.org/10.1016/j.compedu.2006.01.010.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale: Lawrence Erlbaum Associates.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319–340.

Doleck, T., Bazelais, P., & Lemay, D. J. (2017a). Examining CEGEP students' acceptance of CBLEs: A test of acceptance models. *Education & Information Technologies, 22*(5), 2523–2543. https://doi.org/10.1007/s10639-016-9559-9.

Doleck, T., Bazelais, P., & Lemay, D. J. (2017b). The role of behavioral expectations in technology Acceptance: A CEGEP Case Study. *Journal of Computing in Higher Education*. Advance online publication. https://doi.org/10.1007/s12528-017-9158-9.

Enström, E., Kreitz, G., Niemelä, F., Söderman, P., & Kann, V. (2011). Five years withKattis—using an automated assessment system in teaching. In *Proceedings of the Frontiers in Education Conference* (pp. T3J-1-T3J-6). Los Alamitos, CA: IEEE.

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review, 100*, 363–406.

Fernandez Aleman, J. (2011). Automated assessment in a programming tools course. *IEEE Transactions on Education, 54*(4), 576–581. https://doi.org/10.1109/te.2010.2098442.

Fornell, C., & Larcker, D. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18*(1), 39–50.

Garcia-Mateos, G., & Fernandez-Aleman, J. L. (2009). Make learning fun with programming contests. In Z. Pan, A. D. Cheok, W. Müller, & A. E. Rhalibi (Eds.), *Transactions on edutainment II* (pp. 246–257). Berlin Heidelberg: Springer-Verlag.

Hair, J., Ringle, C., & Sarstedt, M. (2011). PLS-SEM: Indeed a silver bullet. *The Journal of Marketing Theory And Practice, 19*(2), 139–152. https://doi.org/10.2753/mtp1069-6679190202.

Henseler, J., Hubona, G., & Ray, P. (2016). Using PLS path modeling in new technology research: Updated guidelines. *Industrial Management & Data Systems, 116*(1), 2–20. https://doi.org/10.1108/imds-09-2015-0382.

Hong, S., Thong, J., & Tam, K. (2006). Understanding continued information technology usage behavior: A comparison of three models in the context of mobile internet. *Decision Support Systems, 42*(3), 1819–1834. https://doi.org/10.1016/j.dss.2006.03.009.

ICPC Kattis (2017). *Icpc.kattis.com*. Retrieved 8 July 2017, from https://icpc.kattis.com/

Jasperson, J., Carter, P. E., & Zmud, R. W. (2005). A comprehensive conceptualization of post-adoptive behaviors associated with information technology enabled work systems. *MIS Quarterly, 29*(3), 525–557.

Kattis (2017). *Kattis.com*. Retrieved 9 July 2017, From http://www.kattis.com/

Open Kattis (2017). Open.kattis.com. Retrieved 9 July 2017, from https://open.kattis.com/

Kock, N. (2015a). WarpPLS. Retrieved from http://www.warppls.com

Kock, N. (2015b). *WarpPLS 5.0 user manual*. ScripWarp Systems. Retrieved from http://cits.tamiu.edu/WarpPLS/UserManual_v_5_0.pdf

Layman, L., Williams, L., & Slaten, K. (2007). Note to self: Make assignments meaningful. In *Proceedings of the 38th SIGCSE technical symposium on computer science education* (pp. 459–463). New York: ACM.

Lee, M. (2010). Explaining and predicting users' continuance intention toward e-learning: An extension of the expectation–confirmation model. *Computers & Education, 54*(2), 506–516. https://doi.org/10.1016/j.compedu.2009.09.002.

Legris, P., Ingham, J., & Collerette, P. (2003). Why do people use information technology? A critical review of the technology acceptance model. *Information Management, 40*(3), 191–204. https://doi.org/10.1016/s0378-7206(01)00143-4.

Lemay, D. J., Doleck, T., & Bazelais, P. (2017). "Passion and concern for privacy" as factors affecting snapchat use: A situated perspective on technology acceptance. *Computers in Human Behavior, 75*, 264–271. https://doi.org/10.1016/j.chb.2017.05.022.

Limayem, M., Hirt, S. G., & Cheung, C. M. K. (2007). How habit limits the predictive power of intention: The case of information systems continuance. *MIS Quarterly, 31*(4), 705–737.

Lin, K. (2011). E-learning continuance intention: Moderating effects of user e-learning experience. *Computers & Education, 56*(2), 515–526. https://doi.org/10.1016/j.compedu.2010.09.017.

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education, 31*(2), 199–218. https://doi.org/10.1080/03075070600572090.

Piech, C., Huang, J., Nguyen, A., Phulsuksombati, M., Sahami, M., & Guibas, L. J. (2015). Learning program Embeddings to propagate feedback on student code. International conference on machine Learning'15, 37, 1093–1102.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137–172. https://doi.org/10.1076/csed.13.2.137.14200.

Shute, V. J. (2008). Focus on formative feedback. *Source: Review of Educational Research, 78228173*(1), 153–189. https://doi.org/10.3102/0034654307313795.

Stone, J., & Madigan, E. (2008). The impact of providing project choices in CS1. *ACM SIGCSE Bulletin, 40*(2), 65–68. https://doi.org/10.1145/1383602.1383637.

Terzis, V., Moridis, C., & Economides, A. (2013). Continuance acceptance of computer based assessment through the integration of user's expectations and perceptions. *Computers & Education, 62*, 50–61. https://doi.org/10.1016/j.compedu.2012.10.018.

Thong, J., Hong, S., & Tam, K. (2006). The effects of post-adoption beliefs on the expectation-confirmation model for information technology continuance. *International Journal of Human-Computer Studies, 64*(9), 799–810. https://doi.org/10.1016/j.ijhcs.2006.05.001.

Venkatesh, V., & Davis, F. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science, 46*(2), 186–204. https://doi.org/10.1287/mnsc.46.2.186.11926.

Wang, T., Su, X., Ma, P., Wang, Y., & Wang, K. (2011). Ability-training-oriented automated assessment in introductory programming course. *Computers & Education, 56*(1), 220–226. https://doi.org/10.1016/j.compedu.2010.08.003.

Wu, C., Gerlach, J., & Young, C. (2007). An empirical analysis of open source software developers' motivations and continuance intentions. *Information Management, 44*(3), 253–262. https://doi.org/10.1016/j.im.2006.12.006.