Engaging Business Students with "Low-Code" Model Driven Development: Self-Efficacy Beliefs in an Introductory MIS Course

Kathie Wright
Salisbury University
kmwright@salisbury.edu

Yvonne L. Antonucci
Widener University
yantonucci@widener.edu

Laura Anderson
Salisbury University
leanderson@salisbury.edu

Andrene Townsend
Widener University
ahtownsend@widener.edu

Abstract

"Low-code/no-code" (LC/NC) platforms are designed to empower non-IT business professionals to use model driven development to rapidly create sophisticated applications. Organizations are increasingly adding LC/NC platforms to their IT software portfolios. Thus, it is likely that current business students will encounter such tools and be expected to be able to use them. This research assesses the implementation of a low-code app development case within a business management information systems (MIS) course to identify whether it promotes student self-efficacy of learning complex technology topics. Statistical analysis of pre and post survey responses indicates that student self-efficacy beliefs increased after completing the case and the change in self-efficacy is positively related to interest in further LC/NC skill development and interest in the MIS major.

Keywords: Model Driven Development, Low-code App Development, Self-Efficacy.

1. Introduction

Technologies are constantly evolving and re-emerging in recombinant and novel forms. This research examines model driven development (MDD) which is an expansion of the Computer Aided Systems Engineering (CASE) tools developed three decades ago. Often characterized as "low-code, no-code" (LC/NC) application development platforms, MDD tools aim to create "citizen developers" by leveraging knowledge worker understanding of business processes. Standard business process modeling notation and team support promote the collaboration of functional business employees with IT staff to increase developer productivity, improve software quality while reducing developer backlogs and maintenance costs, and increase customer satisfaction (Hurlburt, 2021).

Industry analysis of MDD platforms usage across organizations project significant growth in this software development sector (Wong et al., 2021). Enterprise-level providers (Salesforce, Microsoft, SAP) have added MDD development platforms to augment existing services and quickly build applications. Thus, it is likely that current business students will graduate into an environment where they will need to use a MDD approach and feel comfortable using MDD tools in collaborative development projects. While there has been some success of student MDD platform utilization (Henkel & Stirna, 2010), there is little research on success in an entry level MIS course.

This research presents statically analyzed results from an introductory case, "Free as a Bird" (FaaB), using the Mendix MDD platform deployed within an introductory MIS course of a business school. The purpose is to identify whether low-code app development within a MIS course promotes student self-efficacy of learning complex technology topics. It also relates the experience to student interest in further LC/NC development and/or interest in the MIS major.

2. Background

2.1. Model Driven Development and Low-Code/No-Code Platforms

MDD undergirds many commonly used business applications such as Excel and modern web communication platforms (Hurlburt, 2021). Motivated by a need to provide non-IT professionals development-like capabilities, these tools enable business knowledge workers to create technology solutions. MDD platforms are characterized by reusable components and a highly visual interface emphasizing graphical models with "drag and drop" connections between objects to create program workflows.

LC/NC application development tools use similar approaches to both improve ease of use and advance the capabilities of earlier generation programming languages. Automatic page builders containing visual components with automatic backend bindings and extension parameters allow the developer to build the front-end interface quickly. Instead of requiring text-based code, business logic is typically diagrammed

URI: https://hdl.handle.net/10125/103199
978-0-9981331-6-4

using standard modeling notation. Common data connectors enable users to connect to application program interface (API) structures and incorporate data inflows from external sources

Gartner estimates that 70% of new applications developed by enterprises will use low-code or no-code technologies by 2025, up from less than 25% in 2020 (Wong et al., 2021). Organizations increasingly adopt enterprise-level LC/NC platforms to augment existing software development platforms and to leverage non-IT functional talent as "citizen developers", reducing pressures on IT development backlogs and enabling programmers to focus on harder problems.

It also should be noted that employers are resetting degree requirements for IT recruitment in favor of demonstrated skills and competencies (Fuller et al., 2022). Although college degrees continue to be viewed as a proxy for soft skills, inflated degree requirements have been identified as a barrier to successful recruitment. Technical skills validated through pre-employment testing and certifications enable businesses to recruit students who have not yet graduated. This trend, which accelerated through the 2020-2021 COVID-19 pandemic, should be evaluated by educators responsible for modernizing MIS curriculum.

2.2. Application to the MIS Curriculum

By flattening the technology platform learning curve, the MDD approach and LC/NC tools shift application development from a technology-centric focus to one that emphasizes business process, logic, and user design. From a pedagogical perspective, it offers interesting possibilities for presenting and engaging students in MIS topics (Thacker et al., 2021). For example, LC/NC connections between the front-end interface and back-end database tend to be flexible and bi-directional—i.e., interface objects can be bound to existing data attributes, however, it is possible to create new attributes and entities "on the fly" directly from interface page controls. The immediacy of the effect supports conversations regarding the integration of the app interface and data layers and the importance of promoting data integrity.

Similarly, LC/NC program flows are diagrammed visually, often using business process modeling notation (BPMN) as the standard. This supports a common interface for IT and non-IT professionals to collaborate on developing application logic. Within the context of MIS education, this approach enables students to concentrate on interpreting user requirements and mapping corresponding business logic rather than using flowcharts to learn code syntax.

Finally, several LC/NC platforms incorporate agile project/team management and communication tools to support activities surrounding the application development lifecycle. These include technical and cultural DevOps best practices: making customer needs the primary focus, collaborating and sharing responsibility, iterative prototyping and continuous testing, delivery, and deployment. User Experience (UX) and design thinking techniques are explicitly integrated into the development process in the form of user personas, stories, and user testing/feedback mechanisms. LC/NC tools are applicable to several courses in a standard MIS curriculum, including MIS field survey courses, systems analysis and design, project management, and to a lesser degree, programming, and database design (Crumbly & Field, 2020; Litman & Field, 2018; Poe & Mew, 2019). This enables students to experience many of the critical, but less technical MIS roles, such as those offered in UX and project management careers and encourages them to view themselves as future application co-creators.

Given its potential usefulness in addressing industry skills gap and promoting experiential learning across the MIS curriculum, it is particularly important to identify how LC/NC relates to MIS education objectives and how to measure student learning outcomes. This research addresses these areas by posing the following research questions:

(1) How does the use of a LC/NC platform in an introductory MIS survey course affect self-efficacy beliefs? (RQ1)
(2) How does the use of LC/NC tools relate to MIS student learning outcomes? (RQ2)
(3) Does use of a LC/NC platform increase interest in application development and/or MIS major? (RQ3)

2.3. Related Work

Current efforts to evaluate modern LC/NC tools are primarily observational, lacking a consistent theoretical foundation (Thacker et al., 2021). This research utilizes Computer Self-Efficacy (CSE), "an individual's perception of efficacy in performing specific computer-related tasks within the domain of general computing", as a subsidiary construct within social cognitive theory (Marakas et al., 2007). CSE has been shown to enhance individual learning of technology tasks and improve performance outcomes across organizational settings (Karsten et al., 2012; Kher et al., 2013; Stajkovic, et al., 1998). In this research, we use a well-known general CSE scale as well as a second scale specifically germane to use of LC/NC tools to gauge changes in student CSE perceptions after LC/NC training (Marakas et al., 2007).

Studies (Charland et al., 2015; Eder et al., 2019) have adapted the revised Bloom's Taxonomy model (Anderson & Krathwohl, 2001) to relate technology competencies to basic and complex student learning outcomes in business schools. Pre- and post-tests revealed increased knowledge and skills related to application understanding, ability to apply skills, and problem-solving knowledge. Basic knowledge related to Bloom's levels 1-2 (remember, understand), applied knowledge related to level 3 (apply), and problem-solving and using knowledge for decision-making related to levels 4-5 (analyze, evaluate). Here, we extend the mapping of Bloom's model to changes in CSE perceptions of various technology tasks.

3. Methodology

3.1 The Research Project

During the 2020-2021 academic year, significant effort was made to revise the introductory MIS course with the view of incorporating a LC/NC platform (Mendix) into the course curriculum. Mendix was already being used in the application development course and it was determined that students would benefit from an introduction to "no-code" tools earlier in their academic career. Industry partners/advisors were canvased as to the advisability of including LC/NC with a positive response. Finally, an online Mendix case entitled "Free as a Bird" (FaaB) was developed to introduce MIS topics, such as relational database modeling, user design and application development in the context of creating a mobile app.

The FaaB case was designed to expose students to MDD while simultaneously connecting data maintained as worksheet flat files to a relational data model. The entire case utilizes a sophisticated mechanism to parse spreadsheet tables, automatically creating entity associations and user views in the form of application pages. The resulting app is responsive and can be viewed across devices. Because the technical requirements are abstracted, the exercise rapidly moves through the introduction of theoretical concepts such as database relationships, data persistence, application program interface (API) integration and usability.

The case provides step by step instructions for completing the application and discusses data integrity, design thinking and usability. Because the application unfolds in marked stages it is possible to frame the exercise as an iterative process, with natural pauses to consider the app from multiple viewpoints. Subsequent "enhancements" are presented as a set of user stories to mimic agile development sprints.

3.2. Measurement of LC/NC Self-Efficacy Beliefs and Student Learning Outcomes

There is a venerated body of literature linking self-efficacy beliefs to student performance on computer tasks (Compeau & Higgins, 1995; Karsten et al., 2012; Kher et al., 2013).

Table 1. General Computer Self-Efficacy Scale

I could complete the job using the software package:
(1) …if there was no one around to tell me what to do as I go.
(2) …if I had never used a software package like it before
(3) …if I had only the software manuals for reference
(4) …if I had seen someone else using it before trying it myself
(5) …If I could call someone for help if I got stuck
(6) …If someone else had helped me get started
(7) …if I had a lot of time to complete the job for which the software was provided
(8) …if I had just the built- in help facility for assistance
(9) …if someone showed me how to do it first
(10) …if I had used similar packages before this one to do the same job

Table 2. Low-Code Self-Efficacy Measure

I believe I could:
(1) …explain how data is connected in application development.
(2) …explain what responsive views are.
(3) …explain the benefits of low-code development to an end user
(4) …identify data relationships needed for appropriate application development
(5) …identify appropriate data types needed in an application
(6) …develop initial pages of an application
(7) …develop a domain model for the application with appropriate relationships
(8) …change appropriate data types in the domain model
(9) …add values to an enumerated field
(10) …resolve errors in the application development
(11) …identify data relationship issues (referential integrity).
(12) …identify data requiring modification to support application functionality.
(13) …evaluate user interface consistency.
(14) …assess the functionality (usability) of the application.
(15) …conduct tasks related to meaningful evaluation of the application.

Of particular interest is the relationship of self-efficacy to student perseverance and performance.

Increased self-efficacy beliefs promote persistence in the face of challenges. Despite the oft-promoted characterization of LC/NC as a simple development platform, the tools mask considerable complexity which must be understood to use effectively.

To understand the role of self-efficacy on the learning process, a two-pronged framework (Tables 1 and 2) was adopted utilizing constructs from both general and application specific CSE measures (Compeau & Higgins, 1995; Johnson & Marakas, 2000; Marakas et al., 2007). Compeau and Higgins (1995) measure of general computer self-efficacy (GCSE) is a well-established instrument which is purposefully divorced from referencing specific applications (Table 1). As such, it can be used to provide a baseline level of computer technology related confidence.

A second instrument, ("low-code self-efficacy" or LCSE) focusing on application specific attributes of LC/NC was developed following the guidance of Johnson and Marakas (2007). Designed to cohere as a measure of overall LC/NC proficiency, each item references a capability of a LC/NC software. These are presented in terms of progressive complexity to establish a frontier of individual CSE beliefs (Table 2).

Learning outcome measures utilized in this study follows the research of Eder et al. (2019) who mapped student enterprise systems learning outcomes to five of the six learning objective levels in Anderson and Krathwohl's (2001) revised Bloom's Taxonomy. Similarly, the LCSE items were mapped to five of the six learning objective levels in the revised Bloom's Taxonomy model (Anderson & Krathwohl, 2001) (Table 3). Five items addressing students' perceptions of their LC/NC conceptual understanding and ability to explain concepts were aligned to levels 1-2 learning outcomes (remember, understand). Similarly, five LCSE items involving the students' perceived ability to execute low-code application tasks were categorized as a direct measure of the taxonomy level 3 learning outcome (apply). The most complex of the LSCE items were developed to capture students' perceived ability to interpret, evaluate and enhance the LC/NC application usability, resulting in five items mapped to taxonomy levels 4-5 (analyze, evaluate) as evidence of higher order learning outcomes (Table 3).

Table 3. Alignment of Revised Bloom's Taxonomy Levels with LCSE Assessments (Adapted from Anderson and Krathwohl (2001) and Eder et al. (2019))

Learning Objective	Application of LC/NC Knowledge	Assessment
1. Remember *Recall and recognition of information* 2. Understand *Interpreting, summarizing, inferring, comparing, explaining*	Low-Code application basic skills	Students can use their Low-code application development knowledge to understand and explain concepts I believe I could… …Explain how data is connected in application development. … Explain what responsive views are. … Identify data relationships needed for appropriate application development. … Identify appropriate data types needed in an application. … Explain the benefits of low-code development to an end user.
3. Apply *Executing, implementing procedures*	Low-Code application applied skills	Students can develop basic pages and connect data. I believe I could… … Develop initial pages of an application. … Develop a domain model for the application with appropriate relationships. … Change appropriate data types in the domain model. … Resolve errors in the application development. … Add values to an enumerated field.
4. Analyze *Discovery of relationships, differentiating, organizing, attributing* 5. Evaluate *Making judgements based on criteria, checking, critiquing*	Problem-solving/ Decision-making	Students can use, interpret, and evaluate data and errors to fix and enhance application usability I believe I could… … Identify data relationship issues (referential integrity). … Identify data requiring modification to support application functionality. … Evaluate User Interface consistency. … Assess the functionality (usability) of the application. … Conduct tasks related to meaningful evaluation of the application.
6. Create *Plan, produce new ideas, products*	*Not evaluated*	

3.3. Student Intentions Regarding Future LC/NC Development and the MIS Major

One motivation for introducing LC/NC development and creating the FaaB case was to increase the relevancy and appeal of the survey course for introductory MIS students. Accordingly, the two following independent survey items (5-point Likert scale) were proposed: (1) I would be interested in learning how to build other applications in Mendix; (2) I would be interested in pursuing an MIS major. Students were also provided with an open-ended request for comments after completing the exercise.

3.4. Research hypotheses and survey design

A primary objective of this research is to determine whether completion of the FaaB case increases students' self-efficacy beliefs regarding their ability to build an application using LC/NC tools (RQ1). Although related, the GCSE and LCSE scales are designed to measure a general and software specific case respectively and are therefore considered as separate hypotheses:

H1: Students self-efficacy beliefs as measured on the general computer self-efficacy scale will increase post-completion of the FaaB case.

H2: Students self-efficacy beliefs as measured on the LCSE scale will increase post-completion of the FaaB case.

Note that only LCSE items were mapped to the revised Bloom's taxonomy of student learning outcomes (RQ2), hence:

H3: Increases in student self-efficacy beliefs as measured on the LCSE scale will relate positively to corresponding student learning outcomes.

Finally, recall that one of the motivations for developing a LC/NC case and learning module for the MIS introductory course was to increase interest in both application development and the MIS field. This leads to the final two hypotheses (RQ3):

H4: Increases in student self-efficacy beliefs as measured by the LCSE scale are positively related to student interest in learning LC/NC application development (Mendix).

H5: Increases in student self-efficacy beliefs as measured by the LCSE score are positively related to student interest in pursuing and MIS major.

To capture the change in student self-efficacy beliefs and learning outcomes (H1, H2, H3), the research design required paired pre- and post-exercise surveys using the GCSE and LCSE scales. Questions relating to student interest in further LC/NC application development and in the MIS major were included in the post-case survey. Open-ended comments were also solicited on the post-case survey.

3.5. Project implementation and data collection

Implementation of the FaaB case was conducted in business school introductory MIS survey courses at two separate mid-Atlantic universities during the fall 2021 and spring 2022 semesters. Due to COVID-19 pandemic restrictions in the fall semester, all course sections were conducted in a hybrid online format with a mix of Zoom lectures and asynchronously delivered content. In spring 2022, courses returned to an in-person instructor format. For both semesters, the FaaB case was presented over a 3-week time period. During in class periods, instructors would present concepts relating to the corresponding tasks in the case and assist students as they completed the application development steps. For asynchronous delivery, instructors presented the conceptual material via online lecture and offered to assist students needing help via Zoom meetings. Upon completion of the FaaB application, students submitted the app URL for verification and ancillary answer sheets for grading.

To account for possible variation in student backgrounds, demographic characteristics relating to gender, prior course and/or work experience were elicited in the preliminary survey. In addition, because the project spanned multiple instructional settings and alternate delivery modes (arising from COVID-19 concerns) both the pre- and post-case surveys asked the students to identify the course instructor. Students were also asked whether they completed the FaaB case: (1) independently (2) in the classroom with the instructor available; or (3) partially independently with some instructor or other student assistance.

4. Survey results and data analysis

4.1. Demographics

4.1.1. Gender. In the pre-GCSE survey students self-identifying as Male=300 (70%); Female=127 (29.6%); Other=2 (0.4%). A Kruskal-Wallis H test was applied to a pairwise Male/Female analysis ("Other" was too small to pair with either of the other groups). Some significant differences were revealed between males and females (items 1-6) with higher male perceptions of pre-GCSE. It is interesting to note that these differences disappeared across all items in the post-GCSE survey.

An identical analysis of the LSCE pre- and post-surveys yielded similar results. Male perceptions of self-efficacy were significantly higher for all of the "I believe I could…" items except for "…explain what responsive views are" and "…identify data relationships needed for appropriate application development". Again, significant gender differences disappeared in the post-LSCE survey analysis.

4.1.2. Prior programming course experience. Sixty-eight (15.8%) of the students in the survey indicated that they had taken at least one prior programming class. Differences in self-efficacy beliefs for both scales were analyzed using the Mann-Whitney U and Wilcoxon W tests to arrive at a combined Z-score and significance level. These students demonstrated significantly higher self-efficacy beliefs across all items in both scales in the pre-case survey. The proportion of students indicating at least one prior programming course persisted in the post-case survey. Nevertheless, the FaaB case exercise tended to mitigate the differential between the two groups. Experienced students continued to demonstrate significantly higher self-efficacy beliefs on GCSE items 2 and 4 and LCSE items 1, 4, 5, 10 and 12 in the post-case survey.

4.1.3. Course Differences. The FaaB case was implemented across two universities spanning 7 instructors. Five instructors are from one university and two at the other. Kruskal-Wallis H test statistics were generated for both scales with mixed results. For the pre-GCSE scale, items 3,5,6,7,8,9 and 10 were significantly different across instructors with students in courses taught by the two instructors at the second institution demonstrating higher self-efficacy beliefs. There were no significant differences in GCSE beliefs in the post-case survey results. Pre-LCSE survey results did not indicate significant differences, however, post-LCSE beliefs for items 1 and 6 ("explain benefits of low-code", "develop initial pages of app") were significantly higher for one of the instructors at the second university.

COVID-19 course delivery challenges resulted in three possible FaaB case completion modes (independently, with instructor assistance in the classroom, and asynchronously with instructor/student assistance available). Kruskal-Wallis H test results for the three groups did not find significant differences in any of the items in the GCSE and LCSE scales in either the pre- or post-samples.

4.2. GCSE and LCSE construct validity

In order to test the hypotheses, we first assessed the internal consistency of the GCSE and LCSE self-efficacy measures by examining the internal consistency of the items used in each scale (Cerny & Kaiser, 1977).

Analysis of the ten items used to measure GCSE resulted in a Chronbach's alpha score of .940 for the pre-survey and .957 for the post-survey results. Both tests indicate a high level of internal consistency. Further analysis of the inter-item correlation indicates that removal of items 1 and 2 improve the Cronbach's Alpha score from .940 to .948 for the pre-survey. Removal of the same two items from the post-survey results improve Cronbach's Alpha from .957 to .967. Given that the incremental improvement was minimal and the long-standing robustness of the original instrument, we opted to retain all 10 items in the hypothesis tests.

The Kaiser-Meyer-Olkin (KMO) test was applied in the next stage of the analysis to determine the degree of common variance among the 8 items (Kaiser, 1974). The KMO measure for pre-case GCSE is .931 demonstrating a "marvelous" degree of common variance and indicating that a factor analysis will account for a large proportion of the variance. Furthermore, the Bartlett's test of sphericity is less than 0.05 ($p<0.001$), supporting the suitability of factor analysis (Table 4).

Table 4. GCSE Factor Analysis

Factor Analysis Test	Pre-	Post-
Kaiser-Meyer-Olkin Measure of Sampling Adequacy	.931	.910
Bartlett's Test of Sphericity (Chi-Square)	3114.9 ($p<0.001$)	1814.5 ($p<0.001$)

We performed an identical analysis on the pre- and post-survey results for the LCSE measure. For the pre-case survey results, the Chronbach's Alpha reliability statistic was .978 and the corresponding statistic for the LCSE post-case survey results was .987, both indicating a high level of internal consistency. Neither survey statistic was improved by the removal of any of the items, hence all 15 items were retained for the subsequent factor analysis.

The KMO and Bartlett test statistics are presented in Table 5, below. For both the pre-and post-case survey results the KMO test statistic indicates that the degree of common variance among the 15 variables is "marvelous" and that a factor analysis will account for a good amount of variance. Furthermore, Bartlett's test statistic is less than 0.05 ($p<0.001$) significance level indicating responses collected for this study are valid and a factor analysis is suitable (Table 5). A factor analysis of the 15 items used to measure LCSE in the pre-case survey revealed 1 factor with an eigenvalue greater than 1.0, explaining that 75.9% of the common variance can be accounted for by a single factor.

Table 5. LCSE Factor Analysis

	Pre-	Post-
Kaiser-Meyer-Olkin Measure of Sampling Adequacy	.959	.948
Bartlett's Test of Sphericity (Chi-Square)	4234.2 (p<0.001)	3997.4 (p<0.001)

4.3. Hypotheses Testing

A total of 147 paired responses were obtained across pre and post data collection. The paired sample represents approximately 34% of the preliminary survey result (429 responses). The low post-case survey response rate may have been due to the combined effects of COVID-19 and web-based surveys (which were neither mandated nor monitored). Recall that following the relevant computer self-efficacy literature, the GCSE and LCSE used 10 Likert scales for item responses. As the data were not normally distributed, non-parametric tests were used for data comparison (Nachmias & Nachmias, 1987).

Table 6. GCSE paired Wilcoxon Signed Ranks

I could complete the job using the software package…	Z-statistic	Sig. (2-tailed)
(1) …if there was no one around to tell me what to do as I go.	-8.014	<.001
(2) …if I had never used a software package like it before	-6.717	<.001
(3) …if I had only the software manuals for reference	-6.110	<.001
(4) …if I had seen someone else using it before trying it myself	-5.972	<.001
(5) …If I could call someone for help if I got stuck	-3.380	<.001
(6) …If someone else had helped me get started	-2.991	.003
(7) …if I had a lot of time to complete the job for which the software was provided	-3.057	.002
(8) …if I had just the built- in help facility for assistance	-3.988	<.001
(9) …if someone showed me how to do it first	-.257	.797
(10) …if I had used similar packages before this one to do the same job	-.005	.996

Paired responses for both the GCSE and the LCSE scales were evaluated using the Wilcoxon Signed Ranks Test. Negative ranks include paired responses in which the student indicated a higher pre-case self-efficacy response. Positive ranks are indicated when the post-case self-efficacy response is higher, and "ties" reflect no change in self-efficacy beliefs.

With regard to the GCSE scale, the paired student responses for items 1-8 indicated that post-case self-efficacy beliefs were, in the majority of cases, higher, than for the pre-case survey. These results were highly significant. In the case of items 9 and 10, the distribution of negative and positive ranks was less marked and not significant (Table 6).

Table 7. LCSE paired Wilcoxon Signed Ranks

I believe I could…	Z-statistic	Sig. (2-tailed)
(1) …explain how data is connected in application development.	-8.164	<.001
(2) …explain what responsive views are.	-8.758	<.001
(3) …explain the benefits of low-code development to an end user	-8.986	<.001
(4) …identify data relationships needed for appropriate application development	-8.445	<.001
(5) …identify appropriate data types needed in an application	-9.025	<.001
(6) …develop initial pages of an application	-8.964	<.001
(7) …develop a domain model for the application with appropriate relationships	-8.929	<.001
(8) …change appropriate data types in the domain model	-9.163	<.001
(9) …add values to an enumerated field	-8.848	<.001
(10) …resolve errors in the application development	-8.975	<.001
(11) …identify data relationship issues (referential integrity).	-9.059	<.001
(12) …identify data requiring modification to support application functionality.	-9.155	<.001
(13) …evaluate user interface consistency.	-8.721	<.001
(14) …assess the functionality (usability) of the application.	-8.679	<.001
(15) …conduct tasks related to meaningful evaluation of the application.	-8.931	<.001

To test the relationship of the LCSE items to the three Bloom's Revised Taxonomy categories (H3), items were separated into the assigned groupings. Wilcoxon Signed Ranks tests were performed on each group. As shown in Table 8, post>pre-case CSE beliefs were highly significant, supporting H3, although this result should be viewed as somewhat debatable given the LCSE factor analysis. Additional commentary on this issue appears in the Discussion.

Similarly, the results for the LCSE scale also supported H2 in that student post-case low-code self-efficacy beliefs were higher than pre-case beliefs. All items were highly significant ($p<0.001$) (Table 7).

Table 8. LCSE items mapped to Bloom's Revised Taxonomy (paired Wilcoxon Signed Ranks)

Bloom's Revised Taxonomy Categories	Z-statistic	Sig. (2-tailed)
Post-Pre: Basic Skills	-9.400	<.001
Post-Pre: Applied Skills	-9.168	<.001
Post-Pre: Problem Solving Skills	-9.258	<.001

Hypotheses 4 and 5 propose that increases in self-efficacy beliefs subsequent to the FaaB case will be positively related to (1) interest in learning more about low-code (Mendix) application development and (2) interest in pursuing a major in MIS. Student interest is gauged by a single question for each variable on the post-case survey only, measured on a 5-point Likert scale. Again, the Kruskal-Wallis non-parametric test was used to evaluate the relationship of each of the items in the GCSE and LCSE scales with the Mendix/MIS interest responses. Table 9 summarizes relationships of individual items (GCSE & LCSE) to Interest in MIS and Mendix.

Although these results should be interpreted with caution, they do provide evidence that positive changes in self-efficacy beliefs engendered by the FaaB case experience were positively related to interest in both learning more about low-code application development and the MIS major.

These findings are echoed in the open-ended student comments collected in the post-case survey. The comments were largely, but not uniformly positive. Common positive adjectives were "interesting" and "fun", while negative comments often described the case as "tedious" and "hard". The following (positive) comments elaborate on some of the themes motivating this research.

"*This exercise was super eye-opening to app development without actual coding. I have a better understanding of data models and how they are used in app development.*".

"*This exercise helped me understand more in-depth the issues associated with useability of software applications and how to properly approach, assess, and explain them to both programmers and non-programmers.*"

"*This not only showed how to identify the components necessary to build an app, but to also make an app more user-friendly. Operability does not always mean usability. I appreciate this resource because I may have to build my own app one day!*"

Table 9. Summary of GCSE/LCSE items→Mendix/MIS measures

Interest in LC/NC application development (Mendix) • GCSE Pre-: All items are positively related to interest in Mendix. Items 1-7 are highly significant (p<0.01; items 8-10 are significant (p<0.05). Post-: All items are positively related and highly significant. • LCSE Pre-: Items 3 and 5 are positively related and significant; items 4 and 9 are positively related and highly significant. Post-: All items are positively related and highly significant.
Interest in pursuing an MIS major • GCSE Pre-: Item 7 is positively related and significant (.031). Post-: All items are positively related and highly significant. • LCSE Pre-: None of the item relationships are significant. Post-: All items are positively related and highly significant.

5. Discussion

A summary of the hypothesis test results (Table 10) indicates that hypotheses H2, H3, H4 and H5 were fully supported. The single exception was H1 which showed an increase in general computer self-efficacy beliefs as measured by all but items 9 and 10. We offer two possible explanations. First, because the scale focus is *general* CSE, the responses may reflect student responses across varying software experiences. Second, both items require prior task experience which was not available in this training format.

The domain and task specific (LCSE) scale used in H2, however, does provide strong support for RQ1, supporting increased self-efficacy beliefs post-completion of the FaaB case. The mapping of the LCSE items to the revised Bloom's taxonomy (RQ2) was also supported by the H3 results. Finally, positive increases in CSE beliefs were also associated with increased interest in both application development and the MIS field more generally (RQ3).

Table 10. Summary of hypothesis test results

Hypothesis	Result
H1: Self-efficacy beliefs measured by GCSE increase post-case.	Partially Supported
H2: Self-efficacy beliefs measured by LCSE increase post-case.	Supported
H3: LCSE self-efficacy beliefs, categorized by Bloom's Revised Taxonomy increase post-case.	Supported
H4: Changes in Self-efficacy beliefs measured by GCSE are positively related to interest in learning low-code application development (Mendix).	Supported
H5: Changes in Self-efficacy beliefs measured by GCSE are positively related to interest in pursuing an MIS major.	Supported

As explained previously, we consider this research to be exploratory in nature as it was difficult to maintain a consistent research environment. LC/NC platforms are intentionally dynamic; indeed, Mendix releases updated versions of its platform on a two-week cycle. Six of the seven instructors had minimal LC/NC experience prior to implementing the FaaB case, which may explain why student perceptions of two post-LCSE items were higher for this instructor. Additional research is needed. Finally, the research project coincided with highly localized COVID-19 teaching restrictions, resulting in multiple case delivery modes.

Thus, the results were quite gratifying, particularly in their robustness across different universities, course sections and teaching modalities. A key takeaway is the support for H2, validating the creation and use of a software specific self-efficacy measure (LCSE). A second notable finding was the positive relationship between increased self-efficacy beliefs relating to LC/NC and an increased interest in both learning more application development and pursuing an MIS major. Students were clearly able to differentiate between the two objectives; initial student perceptions of their ability to complete LC/NC tasks was higher, perhaps reflecting their familiarity with apps as digital natives and indicating their enthusiasm for learning about app development. It is interesting to note that the same student's initial self-efficacy beliefs did not relate to an interest in the MIS major, but that the relationships were positive and highly significant in the post-case survey. These results suggest that students not only find value in learning about LC/NC app development, but that this possibly increases their understanding of and interest in the MIS field.

There were, of course, research limitations requiring additional thought and careful consideration in subsequent research. Hypothesis 3, while supported, poses questions regarding the categorization of the LCSE items given that they loaded onto a single factor in the preliminary analysis. This could be due to the internal complexity of each LCSE item which may contain aspects relating to multiple categories.

A second possible research weakness regards the measurement of LC/NC (Mendix) and MIS major interest. While the simplicity of a single question is appealing for exploratory research, it could be useful to refine the gauge of student interest with additional questions such as "I can see myself working in the MIS field in the future". Similarly, interest in application development could be expanded to further develop sub-areas, such as user design and/or data modeling.

The final research limitation regards the long-standing theoretical relationship between self-efficacy beliefs, perseverance, and performance. These relationships have been shown to be positive, significant, and robust across a variety of human activities. Although the FaaB case instructions included some individual latitude, such as theme customization, the steps were fairly concrete with the end objective of creating a standard application.

Independent development tasks (e.g., adding a map object) "on your own" could be added to the FaaB case as either a case requirement or an extra-credit activity. Such tasks provide an opportunity to test student perseverance in the face of challenges and to gauge student skill-acquisition (performance). Further investigation is needed for gender, prior programming course exposure and instructor differences. Gender differences in pre- and post- GCSE and LSCE items all indicated males having a higher self-efficacy before completing the FaaB case than females, however there were no significances after completing the FaaB case. Perhaps the engagement in an actual app development mitigated any prior perceptions. Similarly, students with prior programming course experience reveal higher self-efficacy perceptions for all pre-GCSE and -LCSE items, yet after completing the FaaB case there remained a few differences indicating a need to further investigate prior programming exposure. Instructor differences found in pre-GSCE were also mitigated in the post-GSCE, possibly reflecting the general nature of the CSE scale resulting in student responses of varying software experiences.

6. Conclusion

While the adoption of LC/NC tools by businesses is still in a nascent stage, analysis of software trends

(Wong et al., 2021) and recent hiring practices indicate that the movement towards cultivating "citizen developers" is gaining momentum. Our research indicates that business students are enthusiastic, but initially apprehensive about acquiring LC/NC skills. Introductory curriculum, such as the "Free as a Bird" case using Mendix can play a helpful role, increasing self-efficacy beliefs and encouraging students to consider pursuing further skill development. It may also enlarge student understanding of the MIS field in general, prompting reexamination of the major as an important resource for pursing a future MIS related career.

7. References

Anderson, L.W., & Krathwohl, D.R. (2001). *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy*. New York: Longman Publishing.

Cerny, B.A., & Kaiser, H.F. (1977). A Study of a Measure of Sampling Adequacy for Factor-Analytic Correlation Matrices, *Multivariate Behavioral Research*, 12(1),43-47.

Charland, P., Allaire-Duquette, G., Léger, P. M., & Gigras, G. (2015). Developing and Assessing ERP Competencies: Basic and Complex Knowledge. *Journal of Computer Information Systems*, 56(1), 31-39.

Compeau, D. R., and Higgins, C.A. (1995). Computer Self-Efficacy: Development of a Measure and Initial Test, *MIS Quarterly*, 19(2),189-211. https://doi.org/10.2307/249688

Crumbly, J., & Field, D. (2020). Implementing No/Low Code Development in a Systems Analysis and Design Course. *Journal of Strategic Innovation and Sustainability*, 15(5), 116-130. https://doi.org/10.33423/jsis.v15i5.3591

Eder, L.B., Antonucci, Y.L. & Monk, E.F. (2019). Developing a Framework to Understand Student Engagement, Team Dynamics and Learning Outcomes Using ERPsim, *Journal of Information Systems Education*, 30(2), 127-139. https://aisel.aisnet.org/jise/vol30/iss2/6

Fuller, J., Langer, C., & Sigelman, M. (2022). Skills-Based Hiring Is on the Rise. *Harvard Business Review*. https://hbr.org/2022/02/skills-based-hiring-is-on-the-rise

Henkel, M., & Stirna, J. (2010). Pondering on the Key Functionality of Model Driven Development Tools: The Case of Mendix. In: Forbrig, P. Gunther, H. (eds) Perspectives in Business Informatics Research, vol 64. Springer, Berlin, Heidelberg.

Hurlburt, G. (2021). Low-Code, No-Code, What's Under the Hood? *IT Professional*, 23(6), 4-7. https://doi.org/10.1109/MITP.2021.3123415

Johnson, R. D., & Marakas, G. M. (2000). Research Report: The Role of Behavioral Modeling in Computer Skills Acquisition: Toward Refinement of the Model, *Information Systems Research*, 11(4), 402-417. https://doi.org/10.1287/isre.11.4.402.11869

Kaiser, H. F. (1974). An Index of Factorial Simplicity. Psychometrika, 39(1), 31-36.

Karsten, R., Mitra, A., & Schmidt, D. (2012). Computer self-efficacy: A meta-analysis, *Journal of Organizational and End User Computing (JOEUC)*, 24*(*4), 54-80. https://doi.org/10.4018/joeuc.2012100104

Kher, H.V., Downey, J.P., & Monk, E. (2013). A longitudinal examination of computer self-efficacy change trajectories during training, *Computers in Human Behavior,* 29(4), 1816-1824. https://doi.org/10.1016/j.chb.2013.02.022

Litman, M., & Field, D. (2018). Mendix as a solution for present gaps in Computer Programming in Higher Education, (2018). *AMCIS 2018 Proceedings,*14. https://aisel.aisnet.org/amcis2018/Education/Presentations/14

Marakas, G.M., Johnson, R.D., & Clay, P.F. (2007). The Evolving Nature of the Computer Self-Efficacy Construct: An Empirical Investigation of Measurement Construction, Validity, Reliability and Stability Over Time, *Journal of the Association for Information Systems*, 8(1), 16-46. https://aisel.aisnet.org/jais/vol8/iss1/2/

Nachmias, D. & Nachmias, C. (1987). *Research Methods in the Social Sciences* (3rd ed.). New York, NY: St. Martin's Press.

Poe, L.F., & Mew, L. (2019). Implementing Agile as an Instructional Methodology for Low-Code Software Development Courses. 2019 EDSIG Proceedings, Cleveland, OH, 5(4951).

Stajkovic, A., Luthens, F. (1998) Self-efficacy and work related performance: A meta-analysis, *Psychological Bulletin*, 124(2), 240-261. https://doi.org/10.1037/0033-2909.124.2.240

Thacker, D., Berardi, V., Kaur, V., & Blundell, G. (2021). Business Students as Citizen Developers: Assessing Technological Self-Conception and Readiness. *Information Systems Education Journal*, 19(5), 15-30. https://www.researchgate.net/publication/351055548_Business_Students_as_Citizen_Developers_Assessing_Technological_Self-Conception_and_Readiness

Wong, J., Iijima, K., Leow, A., Jain, A., & Vincent, P. (2021). Magic Quadrant for Enterprise Low-Code Application Platforms. https://www.gartner.com/doc/reprints?id=1-275QSBDL&ct=210813&st=sb