Changes In Students' Perceptions Of Programming Skills Using Class Projects

Lori Baker-Eveleth
Robert W. Stone
Norman Pendegraft

University of Idaho

Recent college graduates face challenges when entering the workforce related to confidence with their job-related skills. An application and test of self-efficacy theory using information systems (IS) projects and students' perceptions of their programming skills resulted in students' positively changing their perceptions of their programming ability. The perceptions were impacted by changes in intellectual interest and encouragement of peers and mediated by changes in outcome expectancy. Additionally, changes in self-efficacy impacted changes in perceptions of programming skills mediated by changes in outcome expectancy. The empirical results validate, at least partially, self-efficacy theory in the context of programming and IS projects. Results are discussed along with conclusions and directions for future research.

A variety of challenges exist for new entrants to the work force. Individuals who have recently completed an undergraduate degree should have a skill set allowing them to successfully transition into the work environment. An important part of this transition involving students' skills is the individual's confidence in their ability to successfully use these skills. This confidence can be captured by student's self-efficacy regarding their job-related skills. In general, self-efficacy is the individual's perception of possessing the requisite abilities to successfully perform a specific task (Bandura, 1977; 1982; 1986). A companion to self-efficacy is outcome expectancy which reflects the individual's perception regarding the result or gain from successful completion of these tasks. Expectancies have significant influence in a variety of settings involving behavioral and affective outcomes (Henry & Stone, 2001).

In order to improve a student's transition from the university environment to the work force may depend on an understanding of self-efficacy and outcome expectancy (Jenkins & Garvey, 2001). The research presented below focuses on improving this understanding by applying self-efficacy theory to the development of undergraduate skills in a technical, skill-dependent discipline, information systems (IS). Specifically, the research uses a theoretical foundation of self-efficacy theory to study the development of programming skills which, for undergraduate students majoring in IS is a job-related skill. Furthermore, a unique aspect of this study is the examination of changes within the context of the self-efficacy model from the beginning to the completion of a project requiring programming. In this way, some insights into the development of these expectancies over time may be gained.

For recent graduates, a strong foundation in logic and programming allows migration and adaptation of their IS skills to a variety of business contexts. Encouraging students to continue developing their logic and programming skills necessitates faculty to consider changing the IS curriculum. IS faculty, at one university, undertook this challenge and developed an approach to integrate logic and programming skills across the curriculum. Students were required to complete projects in six IS courses (i.e., four required classes and two electives) with an objective of developing their programming and logic skills beyond the required programming course. The specific objective was to reinforce and improve students' knowledge of programming and logic as the student progresses through the curriculum rather than compartmentalizing these efforts in the required programming class. This setting provides test conditions in which to examine and study self-efficacy and its changes in a job skill environment.

The remainder of this paper is organized into sections presenting the details of this examination of self-efficacy theory and its impact on students' perceptions of their programming skills. First, a theoretical model based on self-efficacy theory is discussed. Second, the classes using these programming assignments and the details of the assignment are described. The hypotheses to be tested are derived from the theoretical framework and the project specifics are stated. An empirical study using a survey of students who have completed at least one of the classes requiring these projects is presented. Finally, the results are discussed followed by conclusions and directions for future research.

The Theoretical Model

The model used to guide the development of a foundation of programming and logic skills in the IS curriculum is rooted in self-efficacy theory. The theory (Bandura, 1986; Bandura, 1982) links an individual's cognitive state to a variety of affective and behavioral outcomes and perceptions of future outcomes (i.e., loss of control, low self-confidence, low achievement motivation) (Staples, Hulland, & Higgins, 1998). Self-efficacy theory has in the past been used to explain user reactions to information technologies (Bandura, 1986; Baronas & Louis, 1988; Hasan, 2003; Havelka, 2003; Martinko, Henry, & Zmud, 1996; Meier, 1985; Potosky, 2002), inclusive of programming and logic skills. Recent research on computer self-efficacy investigated demographic predictors' (e.g., academic major, gender, computer-related experience) influence on business students' self-efficacy (Havelka, 2003). Significant differences in self-efficacy ratings resulted for IS and economics majors compared to management majors as well as those who have greater than five years experience working with computers. Gender differences did not result in any different self-efficacy ratings.

According to self-efficacy theory, expectations (e.g., motivation, performance, and feelings of frustration associated with repeated failure) in large part determine affect and behavioral reactions in numerous situations. Bandura (1986) separated expectations into two distinct types, self-efficacy and outcome expectancy. An individual's belief that he or she possesses the skills and abilities to successfully accomplish a specific task represents self-efficacy. In addition, an individual's persistence to learn a task impacts his or her perceptions of future outcomes influencing their self-efficacy. Outcome expectancy is an individual's belief that by accomplishing a task, a desired outcome is attained. Outcome expectancy is the consequence of the act and not the act itself. Self-efficacy and outcome expectancy have separate impacts on behavior and affect. However, self-efficacy typically has a larger effect than outcome expectancy (Bandura, 1986) and generally self-efficacy has a direct impact on outcome expectancy (Stone & Henry, 2003).

Four groups of variables or experiences identified by Bandura (1977) impact an individual's expectancy evaluations of a specific task. The strongest is the individual's personal mastery or accomplishments regarding the task. Prior success at performing a task increases self-efficacy and outcome expectancy of that task. On the other hand, failing repeatedly at performing a task lowers these expectations (Gist & Mitchell, 1992). In the context of programming skills, experiences focusing on installation and repairs of computer hardware and software as well as experiences solving computer problems could be viewed as appropriate personal mastery experiences (Coffin & MacIntyre, 1999; Hasan, 2003).

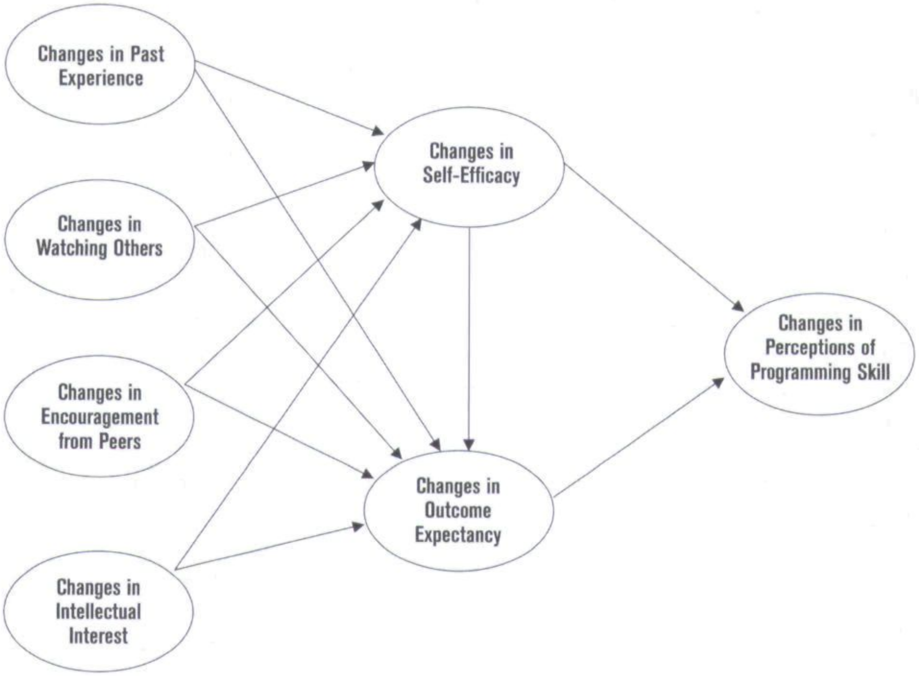
Vicarious experience or modeling the behavior of others who successfully completed the task is the second group of variables. The observer can improve his or her own performance through observing others successfully completing the task (Bandura, 1977; Gist & Mitchell, 1992). In this study, vicarious experience can be viewed as watching teammates or others as they work on similar programming related projects.

Social persuasion is the third group of antecedents to self-efficacy and outcome expectancy. Social persuasion occurs when individuals are led or have it suggested to them that they can successfully complete the task in question and experience the resulting outcome. Common forms of social persuasion are verbal encouragement, coaching, and providing performance feedback (Bandura, 1977). In this study, one form of social persuasion is the encouragement from peers, other students, and faculty to successfully complete a project requiring programming.

The last group of antecedent variables is physiological arousal and emotional states. From here after, this will be referred to as physiological arousal. Physiological arousal of the individual impacts his or her expectancy judgments regarding specific tasks (Bandura, 1977). Improving perceptions of self-efficacy and the value of completing the task occur when there is intellectual interest in a task. Negative judgments of one's efficacy and the task outcome can be produced from anxiety regarding a specific task (Bandura, 1986). For the study at hand, the intellectual interest and stimulation of students regarding a project requiring programming efforts can impact the individual's self-efficacy and outcome expectancy about one's programming skills.

Based on the literature presented above, a model was developed. This model relates the antecedents of self-efficacy and outcome expectancy to students' perceptions of their programming skills, mediated by self-efficacy and outcome expectancy. Because the focus of the research is the changes in students' perceptions of their programming skills due to changes in the model's variables as a result of completing IS course projects, the model is presented in this differential format.

Figure 1: *The Theoretical Model*



The Classes and IS Project Assignments

There are six courses using IS project assignments requiring programming and logic skills. These courses are described below, the first of which is Introductory Systems Development. This course is the required programming course in the major and is the first course in the major completed by students. It is taught at the sophomore level. The current programming language used is Visual Basic.Net. Numerous programming assignments are required in the course beginning with simple form development and culminating with the development of a program interfacing with data in a database.

The second course is a required, junior level course called Modern Information Technology. The course presents the fundamentals of computer hardware and operating systems. In this course, the required IS project develops a simulation of central processing unit (CPU) operations. The simulation is developed in Visual Basic.Net and allows the user to step through a simple program. The simulation must allow the user to initiate the fetch cycle and execute cycle for each instruction in the program and to observe how the register values change.

Systems Analysis and Design is the third required course using IS projects. It is a one semester systems analysis and design course with a greater emphasis on analysis than design. The course is taught at the junior level. The students in this course complete a project based on a case developed by the instructor. Required deliverables are models from the analysis and design of the system based on the case. The last installments of the project require students to design and build a database in Microsoft Access and develop a user interface in Visual Basic.Net to access and update the database.

The fourth class, Business Telecommunications Management is taught at the senior level and is required of IS majors. The project in this course requires students to program a simulation of TCP/IP operations on a network. Specifically, the simulation demonstrates the transmission of a message packet as it walks through the layers between computers. As with the projects in the other courses, the simulation is done using Visual Basic.Net.

The final two courses implementing a required programming project are Database Design and IS Project. Both are senior level courses. Database Design is required in the IS major while the IS Project course is an elective. In Database Design, students develop a trigger and a procedure with a cursor for an Oracle database. The programming component requires students to use PLSQL to accomplish the task. The IS Project course has teams of students performing analysis and design to build a system for a real world client. Because the projects are based on the needs of a client, each project has different programming needs. Typically these projects are extensions or components of a database with a user interface built in Visual Basic.Net.

Hypotheses

The theoretical framework is summarized by hypotheses relating the theoretical model to the empirical study. The model and the hypotheses relating to the theoretical model are displayed in Figure 1. It should be noted that Bandura's original model did not include paths from the antecedents to outcome expectancy as well as the path from self-efficacy to the dependent variable (i.e., perceptions of programming skill). The model used in this study is a variation of Bandura's model that has been previously published in the literature (Stone & Henry, 2003; Henry & Stone, 1995). The theoretical model implies a series of hypotheses presented below.

Hypothesis 1 (H1): Past experience (personal mastery) has a significant and positive impact on self-efficacy.
Hypothesis 2 (H2): Past experience (personal mastery) has a significant and positive impact on outcome expectancy.
Hypothesis 3 (H3): Watching others (vicarious experience) has a significant and positive impact on self-efficacy.
Hypothesis 4 (H4): Watching others (vicarious experience) has a significant and positive impact on outcome expectancy.
Hypothesis 5 (H5): Encouragement of peers (social persuasion) has a significant and positive impact on self-efficacy.
Hypothesis 6 (H6): Encouragement of peers (social persuasion) has a significant and positive impact on outcome expectancy.
Hypothesis 7 (H7): Intellectual interest (physiological arousal) has a significant and positive impact on self-efficacy.
Hypothesis 8 (H8): Intellectual interest (social persuasion) has a significant and positive impact on outcome expectancy.
Hypothesis 9 (H9): Self-efficacy has a significant and positive impact on outcome expectancy.
Hypothesis 10 (H10): Self-efficacy has a significant and positive impact on students' perceptions of programming skills.
Hypothesis 11 (H11): Outcome expectancy has a significant and positive impact on students' perceptions of programming skills.

The Empirical Study

Data collection for the empirical study was conducted by a pre- and post-survey of students during one semester. The six courses with a programming project identified above provided the target population of IS students to survey. Students accessed the survey via a web site prior to working on the assigned IS project and then following the projects' completion. After providing students a statement of informed consent, they were directed to the URL of this site. Participation was completely voluntary, although some incentives were provided to encourage participation depending on the instructor. In some courses, students were given a few minutes of class time to complete the survey whereas in other courses students received two points added to the next in-class quiz if they completed the survey. If participants were enrolled in more than one IS course at a time, they were only able to take the questionnaire for one course. There was a total of 106 IS students eligible to complete the questionnaire. The total response rate for the survey was 78 participants (74% response rate) with 41 responses completing either the pre- or post-survey but not both. Therefore, the number of participants who completed both the pre and post project questionnaire was 37. One explanation for the rather low number of students who completed both questionnaires is a lack of substantial incentive to complete the questionnaire a second time (i.e., after completing the project). In several of the classes, students were given 2 or 3 extra credit points for completing the initial questionnaire. No such extra credit was offered for completing the questionnaire after the project was completed. These observations were used in the empirical analysis presented below. All the statistical analyses were performed in PC SAS version 8.2.

The Sample Characteristics

The summary statistics for the sample are shown in Table 1. The respondents had an average age of 23.6 years and an average GPA of 3.13. Approximately 35 percent of the sample was female. Almost three quarters of the respondents were in either Introductory Systems Development or Systems Analysis and Design when they completed the survey. The percentage of respondents from the remaining courses ranged from 0 to 13 percent. Approximately 38 percent of the sample contained information systems majors without an additional major while another 51 percent were information system majors with a second major in another area. The remaining 11 percent of the sample were majors in Accounting, Computer Science, or undecided as to their major.

Nonresponse Bias

As is the case of any research depending on data collected using a survey, nonresponse is a concern. To examine the possible presence of nonresponse bias, the sample characteristics were compared to the corresponding values of all IS students. IS students enrolled in the College represent the population of students that could have been selected into the sample for this study. The average GPA of all students declared as an IS major within the College was 2.98. Statistical tests indicate that the difference between these two GPAs is not significant at a 20% level in two-tail tests. The average age among all IS students in the College was 21.6 years. Statistical tests show the average ages are not statistically different at a 5% level of significance. IS majors in the College represent 39% of the female population and 61% male. Tests comparing proportions show no significant differences at a 10% significance level.

Also shown in Table 1 are the percentages by major for the College. There are no significant differences using a chi-square goodness of fit test and a 10% level of significance. Based on the comparison of these demographic variables between the sample and IS majors in the College, it is concluded that nonresponse bias does not present a problem.

The Measures

The measures of the constructs in the model were developed in the context of a generic IS project. The generic use of IS projects was necessary since the questionnaire was administered across six different IS courses, each with different requirements for its project. The common thread across the projects in all the classes was development of a functional computer program.

These measures were formed by summing the respondents' answers to selected questionnaire items. The items were summed due to the relatively small sample size of the data set and the estimation technique employed. The individual items are shown in Table 2 grouped in their construct measures. The measures of self-efficacy and outcome expectancy were modified from previously published work by Henry and Stone. Self-efficacy was developed based on Henry and Stone (1995), while outcome

Table 1: *The Sample Characteristics*

Age

Average	Low	High	Number of Observations
23.60	19	48	37

GPA

Average	Low	High	Number of Observations
3.13	2.20	4.00	37

Gender

Gender	Number	Percentage
Female	13	35.14
Male	24	64.86

Enrolled Class

Class	Number	Percentage
Introductory Systems Development	9	24.32
Modern Information Technology	5	13.51
Systems Analysis & Design	18	48.65
Business Telecommunication Management	0	0.00
Database Design	4	10.81
IS Project	1	2.70

Major

Major	Number	Percentage
Accounting & Information Systems	8	21.62
Computer Science	1	2.70
Information Systems	14	37.84
Information Systems & Marketing	2	5.41
Information Systems & Production Operations Management	5	13.51
Computer Science & Information Systems	2	5.41
Information Systems & Finance	2	5.41
Accounting	2	5.41
Other/Undecided	1	2.70

Table 2: *The Items and Summated Measures*

Questionnaire Item	Mean Difference	Min/Max
Self-Efficacy	1.08	-4/13
In general, I am able to successfully complete any IS project I am given.		
I feel more competent working on IS projects than most of my classmates.		
I know enough to successfully complete my IS projects.		
I fully understand how to complete IS projects.		
I am always successful when given IS projects.		
Outcome Expectancy	-0.11	-6/4
Completing IS class projects..........		
Have improved my knowledge of IS concepts.		
Help me learn IS concepts.		
Make it easier to learn IS concepts and skills.		
Past Experience	-0.16	-6/6
I have worked as an IS technician installing hardware, networks, and software.		
I have helped others solve computer problems.		
While not a formal job, I have installed or repaired computer hardware numerous times.		
While not a formal job, I have installed computer software numerous times.		
Watching Others	-0.19	-4/5
Watching and listening to my teammates working on an IS project helps me understand it.		
Watching and listening to other teams working on an IS project helps me understand it.		
Encouragement of Peers	0.38	-4/3
Students who have taken the class help me with the project by encouraging me.		
Friends encourage me about the project.		
Intellectual Interest	-0.19	-6/4
I have found the IS projects required in class to be		
Interesting.		
Challenging.		
Stimulating.		
Perceptions of Programming Skill	0.81	-7/11
I understand the logic related to creating a computer program.		
I can create effective algorithms in any programming language.		
I can explain the difference between inputs, processes, and outputs.		
I understand structured programming.		

expectancy was developed from Henry and Stone (2001; 1995). The measures were modified for the given context of this study, IS projects. For all items, respondents were given a five-point Likert-type scale upon which to respond. The response options provided were strongly agree (5), agree (4), neutral (3), disagree (2), and strongly disagree (1).

Table 2 also displays the descriptive statistics of the measures. The mean differences (pre and post project responses) for the measures range from -0.19 (Watching Others and Intellectual Interest) to 1.08 (Self-Efficacy). Along with these mean differences, the minimum and maximum values for the measures and the number of observations for each measure are displayed.

Other properties of these measures were also examined. The correlations among the measures are displayed in Table 3. The values of these correlations ranged from a high of 0.58 (Outcome Expectancy and Intellectual Interest) to a low, in an absolute value sense of 0.01 (Self-Efficacy and Perceptions of Programming Skill; Self-Efficacy and Encouragement of Peers). The reliabilities of these measures were also examined by calculating the reliability coefficient (i.e., Cronbach's Alpha) for each measure which are also shown in Table 3. The values of these reliability coefficients ranged from a low of 0.62 (Intellectual Interest) to a high of 0.81 (Outcome Expectancy). It should also be noted that several of the measures had rather low reliability coefficients. A potential explanation for this is the rather small size of the sample.

The Estimation of the Model

The estimation of the theoretical model was performed using the mean differences of the summated measures for the 37 paired observations. Each measure represented the construct with the same name in the theoretical model. The estimation approach was a system of simultaneous equations using CALIS (i.e., Covariance Analysis of Linear Structural Equations) in PC SAS version 8. The estimation method used was maximum likelihood.

The fit of the model to the data is summarized in Table 4. The goodness of fit index was 0.96 and adjusted for degrees of freedom it was 0.73. The root mean square residual was 0.05. The chi-square statistic was 5.53 with 4 degrees of freedom which was not significant. The normed chi-square statistic was 1.38. Bentler's comparative fit index was 0.93 and the incremental fit measures ranged from 0.33 to 0.96. These fit measures provide mixed results regarding the fit of the model to the data. The goodness of fit index, root mean square residual, the chi-square statistic, the normed chi-square statistic, Bentler's comparative fit index, and the Bollen non-normed fit index all indicate an acceptable fit (Hair, Anderson, Tatham, & Black, 1992). The remaining incremental fit indexes and adjusted goodness of fit index do not indicate an acceptable fit.

The estimates for the paths in the model are shown in Figure 2. All the path coefficients are reported as standardized values. There are several groups of results regarding these significant path coefficients. These are reported and discussed below.

The first paths discussed are those with significant, positive impacts on students' perceptions of programming skills. Changes in intellectual interest from completing the IS project lead to positive changes in outcome expectancy. Similarly, changes in

Table 3: *The Correlations and Reliability Coefficients*

Correlation Coefficients

Changes in	Self-Efficacy	Outcome Expectancy	Past Experience	Intellectual Interest	Encouragement of Peers	Watching Others	Perceptions of Programming Skills
Self-Efficacy	1.00						
Outcome Expectancy	0.34	1.00					
Past Experience	-0.03	0.19	1.00				
Intellectual Interest	0.24	0.58	0.24	1.00			
Encouragement of Peers	-0.01	0.37	0.03	0.14	1.00		
Watching Others	0.14	0.10	0.30	0.18	0.17	1.00	
Perceptions of Programming Skills	-0.01	0.29	0.25	-0.04	0.13	-0.06	1.00

Reliability Coefficients

Changes in	Cronbach's Alpha
Self-Efficacy	0.69
Outcome Expectancy	0.81
Past Experience	0.66
Intellectual Interest	0.62
Encouragement of Peers	0.69
Watching Others	0.72
Perceptions of Programming Skills	0.65

encouragement from peers also lead to positive changes in outcome expectancy. Changes in self-efficacy from completing the IS project had meaningful positive impacts on outcome expectancy. It is also the case that changes in outcome expectancy produced changes in students' perceptions of their programming skills.

The antecedents of changes in self-efficacy and outcome expectancy were also allowed to pair-wise correlate. One pair of these variables, changes in watching others (i.e., vicarious experiences) and changes in past experience (i.e., personal mastery) was statistically significant. All the estimated correlations among the antecedents are shown in Table 5.

Table 4: *The Summary Statistics of The Model's Fit*

Statistic	Value
Goodness of Fit Index	0.96
Adjusted Goodness of Fit Index	0.73
Root Mean Square Residual	0.05
Chi-Square Statistic	5.53
Degrees of freedom	4
Normed Chi-Square Statistic	1.38
Bentler's Comparative Fit Index	0.93
Bentler & Bonett's Non-Normed Fit Index	0.64
Bentler & Bonett's Normed Fit Index	0.87
Bollen Normed Fit Index	0.33
Bollen Non-Normed Fit Index	0.96

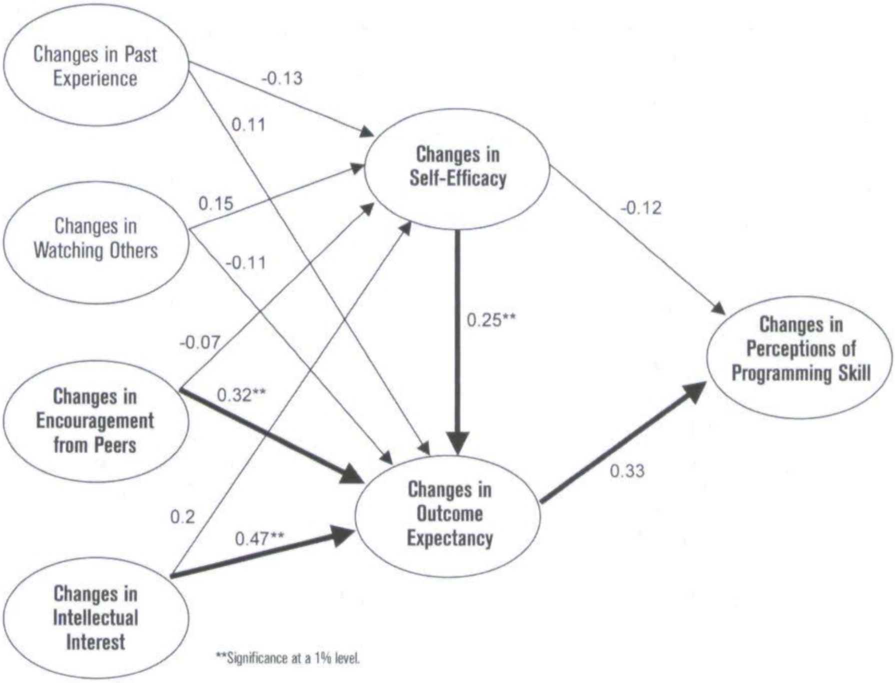
Discussion of the Results

The presented study was designed to examine changes in perceptions of a job-related skill based on changes in the self-efficacy model. The specific environment was students completing IS projects requiring the use of programming skills. The job-related skill examined was students' perceptions of their programming skill before and after completing the class project. The empirical results confirm at least part of what was expected. Completing IS projects improve students' perceptions of their programming skills (i.e., job-related skill) through improving their outcome expectancy (H11). The desired outcome was to improve students' actual programming skills and their appropriate perceptions of this job-related skill. The completion of the project improved student's perceptions of job-related knowledge and skills which significantly influenced perceptions of their programming skills. This result reinforces adding projects with programming components to a series of courses in the IS curriculum.

Three variables in the model had significant impacts on changes in outcome expectancy and ultimately changes in perceptions of programming skills. These variables were changes in self-efficacy (H9), changes in encouragement from peers (H6), and changes in intellectual interest (H8).

Completing the project created changes in self-efficacy which positively impacted perceptions of programming skills mediated by changes in outcome expectancy. An interesting result is that changes in the antecedents to self-efficacy had no impact on changes of self-efficacy. The implication is that the experience of completing the project positively influences self-efficacy, but this change is not explained in the model. Intuitively, it would seem that this influence would be captured by the past

Figure 2: *The Estimated Model Using Standardized Path Coefficients*



experience measure. A potential explanation for not observing this relationship is that students did not link project completion to past experience. The wording of the individual items forming this measure makes this a distinct possibility.

Changes in intellectual interest in the IS project were found to positively impact changes in outcome expectancy. These changes ultimately impacted the perceptions of students' programming skills. The implication of this result lies in the subject and design of the class programming project. If, after completing the project, students find these projects more interesting, stimulating, meaningful and challenging than before completing the project, it positively impacts their perceptions of programming skills through changes in outcome expectancy. This provides instructors opportunities to influence their students through the careful and creative design of projects.

Similarly, changes in the encouragement of peers positively impacted changes in students' perceptions of programming skills, mediated by changes in outcome expectancy. It appears by completing the project, the encouragement from peers convinced students of the value of the project thereby positively impacting their outcome expectancy. Outcome expectancy therefore had a positive impact on students' perceptions of their programming skills.

The measures of changes in past experience and changes in watching others did not have meaningful impacts in changes in outcome expectancy or perceptions of programming skills. As discussed in regard to the similar results for changes in self-

Table 5: *The Correlations Among the Antecendents of Self-Efficacy and Outcome Expectancy from the Model's Estimation*

Antecedent Pair	Correlation
Past Experience and Intellectual Interest	0.24
Past Experience and Encouragement of Peers	0.03
Intellectual Interest and Encouragement of Peers	0.14
Past Experience and Watching Others	0.30**
Intellectual Interest and Watching Others	0.18
Encouragement of Peers and Watching Others	0.17

** Denotes significance at a 1% level.

efficacy, students probably did not link project completion to past experience. It also appears that watching others work on the project does not impact changes in either expectancy or perceptions of programming skills.

Conclusions and Directions for Future Research

Tentative conclusions can be drawn from the empirical results of this study. The meaningful changes in intellectual interest of projects and peer encouragement over the course of project completion have impacted students' perceptions of their programming skills. Furthermore, completing the IS projects has produced positive changes on self-efficacy which ultimately positively impacts changes in perceptions of programming skills. All these impacts are mediated by outcome expectancy. These results support the authors' motivation for requiring IS projects with programming content across multiple courses in the curriculum. The hope was to improve students' perceptions of their programming skills and to help address the "crisis of confidence" observed in many students nearing graduation. These results are sufficiently encouraging to continue with IS projects across the curriculum and to expend effort modifying the projects.

The first of these modifications is to change the projects to help students better make connections between completing the project and improvement in their IS skills and knowledge, including programming skills. In other words, placing the completion of the project in the broader perspective of how completing the IS project improves students' IS skills. One way to provide these linkages is through faculty discussion and persuasion regarding the project and these linkages. A potentially more effective approach would be to use recent graduates as class speakers to help make these linkages. Additional modifications, i.e., questionnaire items, would focus on the discovery of why changes in self-efficacy did not influence changes in perceptions of programming skills. Future research is also needed to determine what factors influence changes in self-efficacy in a significant fashion.

In the future, data on the implementation of the changes discussed above will be collected similar to this study. As an extension of this study, comparison of pre-change and post-change will also include changes to the project, student team, and faculty interaction with student teams. Using this method, the impact from these changes can be studied. As each course builds on a foundation from the preceding courses, projects also become a scaffold. Future research should look at the degree of scaffolding occurring in IS courses and the effect on scaffolded projects. Implicit in these future research studies is the longitudinal nature of this research. Studying the longitudinal data will hopefully lead to answers regarding the role of self-efficacy in impacting changes in the perceptions of programming skills as well as the antecedents to these self-efficacy changes. Data will be collected and analyzed over time to continually evaluate the theoretical model. In addition, measuring changes in actual programming skills rather than perceptions of these changes is being considered. The limiting issue is how to measure actual programming skills. Consideration is being given to a standard exam administered before and after the project's completion.

References

Bandura, A. (1977). Self-Efficacy: toward a unifying theory of behavioral change. *Psychological Review, 84*(2), 191-215.

Bandura, A. (1982). Self-Efficacy mechanism in human agency. *American Psychologist, 37*, 122-147.

Bandura, A. (1986). *Social foundation of thought and action: A social cognitive theory*, New Jersey: Prentice-Hall, Inc.

Baronas, A.K. & Louis, M.R., (1988). Restoring a sense of control during implementation: How user involvement leads to system acceptance. *MIS Quarterly, 12*(1), 111-123.

Coffin, R. J., & MacIntyre, P. D. (1999). Motivational Influences on Computer-Related Affective States. *Computers in Human Behavior, 15*(5), 549-569.

Gist, M. E., & Mitchell, T. R. (1992). Self-Efficacy: A theoretical analysis of its determinants and malleability. *Academy of Management Review, 17*(2), 183-211.

Hair, J. Jr., Anderson, R. E., Tatham, R. L., & Black, W. C. (1992). *Multivariate Data Analysis with readings*, MacMillan Publishing Company: New York.

Hasan, B. (2003). The influence of specific computer experiences on computer self-efficacy beliefs. *Computers in Human Behavior, 19*(4), 443-450.

Havelka, D. (2003). Predicting software self-efficacy among business students: A preliminary assessment. *Journal of Information Systems Education, 14*(2), 145.

Henry, J.W. & Stone, R.W. (2001). The Role of Computer Self-Efficacy, Outcome Expectancy, and Attribution Theory in Impacting Computer System Use. *Journal of International Information Management, 10*(1), 1-16.

Henry, J.W. & Stone, R.W. (1995). A structural equation model of job performance using a computer-based order entry system. *Behaviour & Information Technology, 14*(3), 163-173.

Jenkins, A. & Garvey, R. (2001). Developing human potential at work. *Futures, 33*, 461-467.

Martinko, M.J., Henry, J.W., & Zmud, R.W. (1996). An attributional explanation of individual resistance to the introduction of information technologies in the workplace. *Behaviour & Information Technology, 15*(5), 313-330.

Meier, S. T. (1985). Computer aversion. *Computers in Human Behavior, 1*(2), 71-179.

Potosky, D. (2002). A field study of computer efficacy beliefs as an outcome of training: The Role of Computer Playfulness, Computer Knowledge, and Performance During Training," *Computers in Human Behavior, 18*(3), 241-255.

Staples, D. S., Hulland, J. S., & Higgins, C. A. (1998). A Self-Efficacy Theory Explanation for the Management of Remote Workers in Virtual Organizations. *Journal of Computer-Mediated Communication, 3*(4), 1998.

Stone, R.W. & Henry, J.W. (2003). The Roles of Computer Self-Efficacy and Outcome Expectancy in Influencing the Computer End-User's Organizational Commitment. *Journal of End User Computing, 15*(1), 38-53.