Effect of Reward Expectation on Computer Rapid Application Development Tool Performance in Systems Analysis and Design

William H. Gwinn
University of North Carolina – Wilmington
Cameron School of Business
Wilmington, NC 28403-5611
gwinnw@uncwil.edu

Abstract

An earlier study investigated the use of a computer tool in Systems Analysis versus manual analysis techniques. Expected improvement in user performance through the use of the tool versus manual techniques was not realized. Research in multi-tasking environments seems to indicate that the reward or expected reward for the tool user may have more influence on the project outcome then the properties of the development tool.

This paper analyzes the results of an experiment that measured the change in skill level in junior and senior level college students in the Systems Analysis and Design classes during the spring 2001 semester. Each student was tested on the use of the tool, provided with instruction on advanced features and given practice exercises to perform.

Keywords: Rapid application development, systems analysis and design, application development, application design, MS Access

A number of researchers have studied the psychological aspects of performing multiple tasks and how performance may be degraded by the human information processing overload that may result from multi-task processing. "Information resources such as computer systems are typically used in tasks to improve task outcomes . . . such results are not always realized" (Collins, 1993 p. 18). Collins (1993) contends that tasks that have inconsistent information processing requirements will not be able to develop automatic processes. Tasks should be specifically chosen to present a high-order consistency of information processing requirements so that automatic processing can develop. Students should be asked to make highly similar decisions about similar situations if automatic processing development is desired (Collins, 1993). Tasks that do not become automatic take conscious attention and will cause a decrease in performance for individuals faced with multiple tasks (Thorngate, 1976).

Schneider and Fisk (1982) found subjects performing dual task processing experiments could complete multiple tasks without performance degradation if they were able to achieve automatic processing. If the processing had not become automatic significant decrements in performance occurred despite intensive training on the task performance. Collins (1993) reiterated that degradation in dual task performance may be avoided if information technology use can become automatic.

During the testing of an artificial intelligence tool designed to support student systems analysts, the degradation caused by multitasking was observed (Gwinn 2000). The expected performance improvement using a software tool in a second test (2) over an initial test (1) was expected to parallel or exceed manual performance. The manual performance started with a score of 35.9 and improved to 48.4. The tool performance started at 36.5 and decreased to 32.8 as shown in Figure 1 below.

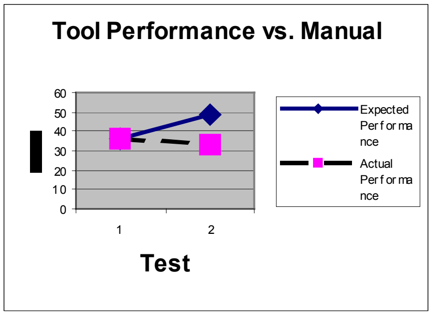

Figure 1. Tool Performance Comparison

The conclusions postulated that the incentive given students in the first experiment was unable to compete with overriding demands of the end-of-semester crunch. The original incentive was to offer every student participating in the experiment five bonus points to be added to their semester grade. The bonus was for participation and not tied to performance. It was speculated that perhaps a change in the incentive to a merit or performance based award of bonus points would make a difference in software tool performance.

This study focused on the reward aspect in testing for mastery of computer software tools as applied to application design. In order to further investigate the influence of the reward system, analysis and design students were given the more familiar Microsoft ACCESS 2000 database to use as a rapid application development tool in their information system analysis and design classes. An experiment was conducted to measure the difference in mastery of the tool at course entry and course exit.

1. EXPERIMENT

A baseline Access skills assessment was administered using the Course Technologies Access Skill Measure online software knowledge test. This was done to establish the degree of familiarity and application knowledge students in Systems Analysis and Design classes retained from their introduction to computing and database courses. During the spring 2001 semester, thirty-six students received instruction in application design using both manual and computer rapid application development techniques. MS ACCESS served as the computerized application prototyping tool. Each Student completed an ACCESS skill test for the beginning, intermediate and advanced level. The three scores were averaged to arrive at the performance score for the student. Bonus points were awarded base on participation + improvement on each skill level. If a student demonstrated improvement on all three levels, the maximum number of bonus points was awarded. The skills tests were administered in a computer classroom to the group as a whole. Each student was allowed 90 minutes to complete all three skills tests. The skill test software automatically recorded each student's score on each test. The tests appeared in basic, intermediate and advanced skill sequence.

Once the initial test scores were established each student received five lectures on using access as a rapid application development prototyping tool. Each student completed a sample exercise and then used ACCESS to complete their term project. After the students completed their analysis and design term projects the Skill Measure software was again administered and the ending level of performance compared to the entry level of performance. Again all three skills levels, basic, intermediate and advanced, were measured. Thirty-two students completed the experiment participating in both entry and exit tests. Two of the original thirty-six students dropped the class and two others were absent on the day the second test was administered. Care was taken to insure the second skill test occurred during the same week at the end of the semester as the earlier experiment to duplicated, as much as possible, the end-of-semester stress and student work load (see Figure 2 below).

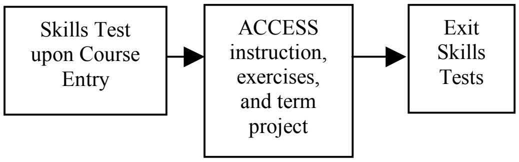
Skills Test upon Course Entry
ACCESS instruction, exercises, and term project
Exit Skills Tests

Figure 2. Experiment Layout

Hypothesis Testing

The experiment was designed to test the null hypothesis that the use of a merit based award of bonus points would not effect the average results scored on the post-test (test 2) and they would be not significantly different from the pre-test (test 1) results. This hypothesis was expressed as:

H_0: The test 2 mean score will be less than or equal to the test 1 mean score.
H_1: The test 2 mean score will be greater than the test 1 mean score.

A significant difference between test results would result in rejection of the null hypothesis H_0.

2. EXPERIMENT RESULTS

The experimental results were analyzed using a paired student's t test. The result of t(0.05, 31) = 8.0076 ; p=2.420E-8 caused the rejection of the null hypothesis:

H_0: The test 2 mean score will be less than or equal to the test 1 mean score.

The test 2 mean score 77.37 was significantly different and greater than the test 1 mean score of 61.77. The initial expectation that the use of a performance based incentive would overcome end-of-semester apathy and reflect the normal positive learning effect from repeating similar tasks over time appeared to be supported by these results.

The skills test recorded scores for the students' basic intermediate and advanced ACCESS skills. The basic and intermediate skills were initially gained in earlier introduction and database courses. The systems analysis course work served to further hone those skills. The advanced skills associated with using macros and visual basic for applications modules to produce custom user applications were taught exclusively in the systems analysis and design course. A second question was generated to look at the advanced skill performance change over the semester. The null hypothesis was that the reward system would have no effect on the advanced level performance difference between test 1 and test 2.

> $H2_0$: There will be no difference in advanced skill performance between test 1 and test 2.
>
> $H2_1$: Performance reflected in test 2 will show an increase over the performance level achieved on test 1.

A significant difference between test results would result in the rejection of the null hypothesis $H2_0$.

The paired student's t test was used to compare the test 1 and test 2 results at the advanced skill level. Hypothesis $H2_0$ was rejected based on the paired student's t test result of: t(0.05, 31) = 7.0351; p= 3.35968E-08. The test 2 average score of 71.8 was a clear improvement over the test 1 average score of 50.25. Improvements in new skills as well as improvements in prior skills were reflected in the experimental results.

3. CONCLUSIONS

This research provides further insight into the impact of a merit rewards system vice a level reward system in student performance and achievement using software tools. The earlier experiment indicated student mastery of software skills but reflected low performance marks versus manual analysis and design skills. Instruction in the use of the tool and practice in tool use did not result in the positive learning curve expected for repeating similar tasks. One of the reasons postulated for this result was that the level reward system used in the earlier experiment could not compete with end-of-semester stress and course demands. The aim of this experiment was to focus on the effect of a merit reward in student performance using a software tool for analysis and design. The improvement reflected in the experimental results was similar to the improvement generalized by the learning curve when repeating similar tasks. It appears this result will be achieved if the reward system can focus or hold the students attention. The original experiment failed in this respect because of a level reward. The students could achieve the reward by simple participation and as a result, placed less emphasis on second test performance then simply completing the task and moving on to another activity with a higher perceived reward. The merit reward based on the level of performance, employed in this experiment, appeared to hold the students' focus on the task at hand resulting in an increase in performance.

4. REFERENCES

Collins, Rosann Webb, 1993, Impact of Information Technology on the Processes and Performance of Knowledge Workers. Doctoral Thesis, Graduate School, University of Minnesota.

Cook, Thomas D. and Donald T. Campbell, 1979, Quasi-Experimentation. Houghton Mifflin, Boston, MA.

Dennis, Alan, and Barbara Haley Wixom, 2000, Systems Analysis and Design. John Wiley & Sons, New York, NY.

Doll, William J., and Gholamreza Torkzadeh, "The Measurement of End-User Computing Satisfaction." MIS Quarterly, 1988, 12.2, pp. 259-274.

Easton, Annette, and George Easton, 1996, Cases for Modern Systems Analysis and Design. Benjamin/Cummings,Menlo Park, CA.

Goodhue, Dale L. "Development and Measurement Validity of a Task-Technology Fit Instrument for User Evaluations of Information Systems." Decision Sciences, 1998, 29.1, pp. 105-138.

Gwinn, William H., "Software Support in the Classroom: Help or HINDRANCE." Proceedings, ISECON 2000, November 2000 Philadelphia, PA

Haag, Stephen, Maeve Cummings, and James Dawkins, 2000, Management Information Systems for the Information Age. 2nd Ed. Irwin McGraw-Hill, Boston, MA.

Hoffer, Jeffery A., Joey F. George, and Joseph S. Valacich, 1999, Modern Systems Analysis and Design. 2nd Ed. Addison-Wesley, Reading, MA.

Keppel, Geoffrey. 1991, Design and Analysis. Prentice Hall, Englewood Cliffs, NJ.

Kendall, Kenneth E. and Julie E. Kendall, 1999, Systems Analysis and Design. 4th Ed. Prentice Hall, Upper Saddle River, NJ.

McHaney, Roger, and Timothy P. Cronan, “Computer Simulation Success: On the Use of the End-User Computing Satisfaction Instrument: A Comment.” Decision Sciences 29.2, 1998, pp. 525-536.

Montgomery, Douglas C. 1991, Design and Analysis of Experiments. 3rd ed. John Wiley & Sons, New York.

Satzinger, John W., Robert B. Jackson, and Stephen D. Burd. 2000, Systems Analysis and Design in a Changing World. Course Technology, Cambridge, MA.

Schneider, Walter and Arthur D. Fisk, “Degree of Consistent Training: Improvements in Search Performance and Automatic Process Development.” Perception & Psychophysics, Vol. 31, No. 2, February 1982, pp. 160-168.

Thorngate, Warren. “Must We Always Think Before We Act?” Personality and Social Psychology Bulletin, Vol. 2, No. 1, Winter 1976, pp. 31-35.