Creating an Interactive Environment for Software Application Learning in Graduate Programs

Sy-Chyi Wang¹ & Jin-Yuan Chern²*

¹ Department of E-learning Design and Management, National Chiayi University, Chiayi County 62103, Taiwan
² Department of Health Care Administration, Chang Jung Christian University, Tainan City 71101, Taiwan

*Corresponding author: Department of Health Care Administration, Chang Jung Christian University, No. 396 Changrong Rd. Sec. 1, Gueiren Dist. Tainan City 71101, Taiwan
Tel. 886-6-2785123 ext 3057 E-mail: chern@mail.cjcu.edu.tw

Received: November 19, 2012 Accepted: December 11, 2012 Online Published: December 23, 2012
doi:10.5430/jct.v2n1p1 URL: http://dx.doi.org/10.5430/jct.v2n1p1

Abstract

Despite the growing application of design and development of constructivist teaching and learning strategies in many fields of study, the effort devoted to exploring the role of the constructivist view of teaching methods for computer software learning is still limited. This study experimented with a new teaching strategy for software applications, in which more interactions and class participation were expected. The strategy had been previously implemented in an undergraduate level course. This was a continuation at the graduate level and aimed to investigate the experiences from both the perspectives of the instructor and students. The findings show that the students strongly felt that the innovative teaching strategy helped them better concentrate in class, which resulted in more peer interactions and class participation. The instructor was satisfied with and appreciated the teaching innovation and considered the strategy effective in meeting the learning goals. Several drawbacks were discussed based on the experience.

Keywords: interactive learning; collaborative learning; teaching innovation; software application; teaching strategy

1. Introduction

There is no doubt that research on using computer technology to enhance teaching and learning is rapidly growing. To meet the computer skill demands in almost every field, schools have been including software application learning in their curriculum. However, compared with the numerous studies on how to enhance teaching and learning with technology, the investigation on the methods and strategies of teaching software applications itself has rarely been conducted. It seems that, in the general view, teaching software application is all about repeated operations and does not have much room for innovative teaching approaches. While several studies introduced some tutorials or process capturing software (such as screen recorder software) for the convenience of self-directed learning on software, the development of classroom teaching methods and the investigation of its effectiveness are lacking.

Some of the few studies indicated that most of the software application instructors used a systematic approach for teaching and considered it most effective (Lambrecht, 1999; McEwen, 1996). Their systematic learning approach was based on a behavioral learning theory assumption, in which learning is regarded as a behavioral response to stimuli (Chen & Ray, 2005). In this approach, instruction usually involves step-by-step directions and teacher-centered methods to complete learning tasks (McEwen, 1996). Lambrecht (1999) concluded that this systematic approach for software teaching is better for novices, but not for advanced learners who seek problem solving abilities and concept transferability in the real world.

In graduate programs, students are in an advanced education environment in which they should be capable of self-discipline and self-direction. Research techniques, decision-making skills, and problem solving in teams are essential skills to be acquired in graduate programs (Campbell, 1996). They should have confidence to work constructively in teams as part of establishing professional skills (Hudson, 2000). They should be able to reach the