Case Report

Enhancing Entrepreneurship and Professionalism in Medical Informatics Instruction: A Collaborative Training Model

CATHRYN A. CARROLL, PhD, BSPharm, WALT RYCHLEWSKI, PhD, MARTY TEAT, MBA, DARRIN CLAWSON, MBA

Abstract This report describes an innovative training program designed to foster entrepreneurship and professionalism in students interested in the field of medical informatics. The course was developed through a private–public interinstitutional collaboration involving four academic institutions, one private firm specializing in health care information management systems, and a philanthropic organization. The program challenged students to serve in multiple roles on multidisciplinary teams and develop an innovative hand-held solution for drug information retrieval. Although the course was technically and behaviorally rigorous and required extensive hands-on experience in a nontraditional learning environment, both students and faculty responded positively.


There is little doubt that medical informatics occupies a central place in modern medicine. Innovative methods for managing health care information are critical to solving the problems posed by our nation’s health care system. However, medical informatics solutions cannot be applied without a well-trained labor force.

Criticisms of the current educational system primarily center on the graduate-level focus of most programs and the lack of an interdisciplinary team approach to educational initiatives. This last point is particularly important given Carter’s astute observation that successful medical informatics applications are the shared responsibility of the medical informatics community, professional societies, academic organizations, and practitioners.

Affiliations of the authors: Division of Pharmacy Practice & Department of Economics, University of Missouri-Kansas City, Kansas City, MO (CAC); Department of Business and Computer Science, William Jewell College, Liberty, MO (WR); Cerner Corporation, North Kansas City, MO (MT, DC).

The authors recognize the generous financial support provided for the development and implementation of the course by the Kauffman Foundation. In addition, the contribution of other faculty members as well as academic administration to the success of the project is also recognized. Faculty recognitions extend to Nathan Beyer (Cerner), David Edwards (Cerner), Gerry Williams (Rockhurst University), Thomas Connolly (Baker University), Eddy Burris (University of Missouri-Kansas City), Susan Santos (University of Alabama-Huntsville), and Judy Mullins (University of Missouri-Kansas City). Lastly, the authors thank the students who participated in this nontraditional educational model. Hopefully, they are better prepared for their first employment experience because of their efforts.

Correspondence and reprints: Cathryn A. Carroll, PhD, BSPharm, Director, Patient Care Services Research, Children’s Mercy Hospitals and Clinics, 2401 Gillham Road, Kansas City, MO 64108; e-mail: ccarroll@cmh.edu.

Received for publication: 09/23/03; accepted for publication: 03/15/04.

These educational issues led the authors to develop an innovative, multidisciplinary undergraduate and professional educational program in medical informatics. The course description and key outcome observations are outlined below.

Case Description

In the spring of 2003, multiple organizations within the greater Kansas City area united to offer an innovative, interdisciplinary training program in medical informatics. Members of four academic institutions and one private organization formed six multidisciplinary student teams that included 34 individuals enrolled in computer and health science disciplines.

Prior to arriving for the first class, students read half of the required course textbook, The Inmates Are Running the Asylum by Alan Cooper. This text is a good training guide for the student without much technology development experience and succinctly highlights historical mistakes frequently made with traditional software development processes.

Students met at the corporate sponsor’s headquarters one evening weekly for four hours of didactic training. Most of the materials (e.g., concepts of software development, requirement gathering, and project management) were provided by the corporate sponsor faculty, and closely replicated the framework and steps involved in that organization’s software development cycle. The training sessions were designed to teach students the general concepts and processes of software development at the knowledge level only. The students were required to operationalize the concepts provided in class using whatever technique their team deemed appropriate. Corporate resources, such as access to software and databases, were available to each team after the members signed confidentiality agreements. Students were expected to spend approximately nine hours per week outside of class on team activities and meetings with their academic faculty advisors.

Each team was composed of a project manager, product manager, application architect, application developer, and
assessment. In general, students overestimated their difference between the peer group ranking and the self-calibration. Performance calibration was defined as the present students with feedback on their performance professionalism and technical performance throughout the semester shaped the ultimate design of the product. Teams researched, identified, and adopted the technology best suited to meet their customers’ needs in the most efficient and effective manner.

As a final examination, each team formally presented their software in competition with the other teams in the class. The teams produced a functioning software solution, a marketing presentation, and a reference document outlining the steps that had occurred through the software development process. The primary evaluation criterion was the congruence between the final solution and the customer requirements. Too often, young entrepreneurs assume that software development success is related to the language, syntax, or quality of the code that is written, when, in reality, the ability to closely match a software solution’s vision, feature/functionality, design, and user interaction to the user’s needs determines success.

Methods and Results

Three evaluation methods were used to assess student and course performance: (1) student peer evaluation and self-assessment to appraise professionalism and technical role performance, (2) final product evaluation to assess entrepreneurship, and (3) course assessments by faculty and students.

Peer Evaluation and Self-assessment

Peer evaluation and self-assessment provided behavioral feedback based on the team’s perceptions of an individual’s professionalism and technical performance throughout the semester. Faculty members used the two assessments to present students with feedback on their “performance calibration.” Performance calibration was defined as “the difference between the peer group ranking and the self-assessment.” In general, students overestimated their behavioral performance relative to peer assessment. In contrast, students generally underestimated their technical performance relative to the peer evaluation.

In addition, the information garnered through these assessments permitted feedback to the student relative to their individual “layoff potential.” For example, students who were considered to be behaviorally strong and technically strong were generally portrayed to have low layoff potential. In contrast, those who were behaviorally weak and technically weak were portrayed to have high layoff potential. At the completion of this analysis, 46% of the students had low layoff potential, 23% had moderate layoff potential, and 31% had high layoff potential. This exercise was intended to better prepare students to be constructive recipients of feedback and help them identify, prior to their first employment experience, those behavioral or technical performance attributes that may negatively impact their future success.

Final Product Evaluation

One of the course’s primary goals was to provide students with the opportunity to experience entrepreneurship through the software development process. Various attributes measured the quality of the alternative competitive solutions that the students created. In general, the competitive solutions were fairly diverse with respect to functionality and appearance. However, there was only a 0.04-point difference in total score between the “winning” solution and its closest competitor. The mean values for each of the performance criteria for all solutions across all teams are presented in Table 1.

Student and Faculty Course Evaluation

Student and faculty assessments included an evaluation consisting of ten questions with each question ranked on a Likert scale ranging from 1 = strongly disagree to 5 = strongly agree. When responding to the question, “Overall my rating of this course is excellent,” the mean student response was 4.67, while the mean faculty response was 4.57.

The mean student ratings of satisfaction with learning goals and class assignments were 4.42 and 4.66, respectively. In addition, students favorably ranked (4.66) teaching methods that allowed them to engage in their own nondirected learning. In the nondirected approach, students experienced the challenges related to balancing lofty end-user expectations, conflicting and changing software requirements, resource constraints, intrateam conflicts, technology limitations, and project management and execution. On the other hand, students reported lower levels of satisfaction (4.16) with the level of faculty interaction outside of class. A final benefit of the course was the opportunity for students to showcase their talents to a potential employer. In fact, at the completion of the course, approximately 30% of the students were hired by the corporate sponsor.

Faculty assessments were also very positive. Faculty members described the course as well organized (4.86). In addition, the faculty believed that the course enhanced their ability to (1) offer more opportunities to students, (2) compare and contrast corporate and academic work environments and cultures, (3) work with colleagues with diverse expertise, and (4) network with peer organizations. Faculty members suggested providing a better context for and framework around the student assessment process for grade assignment (4.29).

Discussion

Interdisciplinary multi-institutional endeavors can be difficult. First, corporate associates must participate in the...
training effort and house the course at the corporate site. These two decisions immediately change the academic culture of the educational experience. First, students take the course more seriously because faculty members may be future employers. Second, the material presented by corporate faculty is considered more credible because the material reflects the “real world.”

The academic institutions also bear a share of the responsibility in making industry practicum models work. Academic leadership must support such endeavors. Faculty members must learn how to teach in ways that go beyond lecturing and testing students. More specifically, interdisciplinary training models require faculty members to manage both the technical and the behavioral aspects of achievement. Students must learn to work in an interdisciplinary fashion, respect the professional diversity of others, and move beyond the traditional training model of “waiting for their assignment.” Having identified key barriers to effective collaborations, it is important to recognize that none is insurmountable and the benefits of the effort clearly exceed the cost.

References