# REAL PROJECTS, REAL LESSONS – WHAT STUDENTS LEARN AND WHAT THE INSTRUCTOR HAS LEARNED

**Ted J. Strickland, Jr., University of Louisville, ted.strickland@louisville.edu**

## ABSTRACT

*Information systems students entering today's workforce are expected to have technology skills, interpersonal skills, and problem solving skills. The senior-level systems project course provides an opportunity to integrate these skills within the context of developing web applications. Experiences designing and instructing this course are described, including descriptions of the course structure and objectives. Learning outcomes are demonstrated. Students' responses to the learning environment indicated that the experience enhanced professional development.*

**Keywords:** interpersonal skills development, information systems curriculum, information systems project development course, physical design and implementation course.

## INTRODUCTION

Which knowledge sets and skills should information systems students possess when they enter the workforce? What do employers expect of today's graduates? How do information systems programs provide the expected type and level of preparation? These fundamental questions are common at information systems department meetings, and they present challenges to faculty members and corporate advisory boards charged with keeping curricula at the forefront of information technology.

Curriculum reform within my department has struggled with the technical elements of the program – which technologies to include, and how to maintain currency among the faculty members teaching these advanced topics. However, technical knowledge is only part of the answer. My corporate advisor panel has reiterated the message that "soft" skills are critical to the success of an information systems professional. Employers seek individuals who think analytically, solve problems within a team environment, communicate effectively, and understand that information technology enables organizations to improve performance.

Employers [6, 7] and educators [1, 2, 3, 4, 5] recognize the need for providing information systems students with opportunities to develop problem solving and interpersonal skills. For example, a 2001 survey of employers, conducted by InformationWeek (see Table I), ranked the importance of problem solving, interpersonal skills, and computer skills [7].

Employers seek well-rounded information systems graduates. As educators, the information systems profession recognizes the importance of developing interpersonal skills. This paper describes an approach to enhancing these skills within the context of the senior-level information systems project course. The presentation includes discussion of the course objectives and structure, an indication that course learning objectives were achieved, and observations regarding student learning during the course.

Table I. What Employers Want – The Most Important Attributes of New IT Employees

| Ranking | Characteristics |
|---|---|
| 1 | Problem-solving skills |
| 2 | Learn new things quickly |
| 3 | Analyze and interpret data |
| 4 | Teamwork skills |
| 5 | Oral-communication skills |

| Ranking | Characteristics |
|---|---|
| 6 | Motivation |
| 7 | Innovative thinking |
| 8 | Self-starter |
| 9 | Written-communication skills |
| 10 | Job-specific computer skills |

## PROJECT COURSE OBJECTIVES, STRUCTURE, AND LEARNING OUTCOMES

The integrative role of an upper division project-oriented implementation course is described in the model curricula for information systems programs [2, 3]. Under IS 2002 guidelines, the course provides a senior-level experience that requires students to apply technical knowledge in a team-based implementation of an information system.

The University of Louisville's metropolitan setting provides access to local business organizations that need help in developing information systems. Business problems have been used in the project course since the mid-1990s. Recently, these projects have featured web applications for management reporting and process improvement. One project involved development of a dynamic report generator for the public school system. Another project created a browser-based user interface and centralized database to automate a manual sales process. These "live" settings create environments where knowledge integration expands beyond the information systems area, and includes multiple elements of business.

Within this context, the project course has evolved as the capstone course in the Computer Information Systems (CIS) program. It aligns closely with IS 2002.9, Physical Design and Implementation in Emerging Environments, but it contains many of the project management elements of IS 2002.10, Project Management and Practice [3]. The course balances development of technical skills with refinement of business awareness and interpersonal skills. The technical elements emphasize design and implementation of a web application. An equal emphasis is placed on the process elements of projects – team building, oral presentations, written communications, leadership and initiative, group decision-making, and project planning and coordination. Opportunities for students to enhance their interpersonal skills are built into every project deliverable. The course learning objectives are as follows:

1. Develop an information system in accordance with analysis and design specifications. Refine analysis and design deliverables to conduct the detailed design and the implementation phases of the system development life cycle.
2. Formulate a proposal and plan of work for conducting project activities, and monitor development progress with respect to that plan.
3. Use prototyping techniques to develop key elements of the information system. Integrate and refine the prototypes to create a functional system.
4. Gain experience with group decision-making and as a member of an information systems development team. Take the initiative in contributing to and in leading group activities.
5. Practice written and oral communication skills.

In designing the project course, the CIS faculty recognized that implementation of an information system is difficult to manage in a single course. Previously, compressing analysis, design, and implementation activities into a single-semester course resulted in incomplete and malformed systems. An "integrated" approach that uses the business problem as the basis for the systems analysis and design course (CIS 320) and the project course (CIS 420) was adopted. Coupling the systems analysis and design course with the project course provided sufficient time for students to analysis, design, and implement meaningful solutions to the business problem.

The business problem was introduced in the systems analysis and design course, CIS 320. As students learn the principles of analysis and design, they applied these concepts to the creation of project deliverables; i.e., an active learning model. During this course, student teams produced deliverables corresponding to the planning, analysis, and logical design phases of the systems development life cycle; e.g., a business case, analysis models, design models, and project dictionary. Each team addressed the same business problem, so that milestone reviews provided an opportunity for student teams to learn from one another. At the end of the semester, each team consolidated its deliverables into a systems design specification.

The project course, CIS 420, continued the systems development life cycle. The latter phases were emphasized – physical design and implementation. A Rapid Application Development (RAD) approach was used to provide students with an iterative perspective on systems development. As a starting point, a system design specification produced in the systems analysis and design course (CIS 320) was selected and provided to each student team. Students were provided a milestone schedule. However, each student team formulated its own project plan for fulfilling the milestones. A proposal and plan described the team's approach. It defined the objectives of each of five prototypes, two design revisions, the user manual, and the project report. A Gantt chart was produced as a means of coordinating team member activities. In this manner, each team proposed a solution to the problem, but each in its own way. Parallel solutions provided the business client with a choice of functional systems. Elements of the course are summarized in Table II.

Table II. Project Course Elements

| Practice | Description / Comments |
|---|---|
| Use of RAD | "Grow" the software as a series of five prototypes, reviewed at regular intervals throughout the semester. Prototyping revealed requirements and design decisions that were omitted during analysis. |
| Design Reviews | Two formal reviews were intermixed with the prototyping schedule. The reviews forced revision of the systems design specification to document requirements and design decisions resulting from prototyping. |
| Project Report | Production software, user manual, and system design documentation. The report consolidated project deliverables into a form suitable for system installation and support by the business client. |
| Attendance of Client at Milestone Reviews | Milestone reviews allowed teams to evaluate the merits of its approach and deliverables in relation to those of "competing" teams. Feedback from the business client helped students understand the importance of providing a solution that meets the business needs – technically, organizationally, and economically. |
| Student Monitoring | Weekly progress reports and periodic peer evaluations. The progress reports focused on individual activities and contributions for the current week and upcoming week, as well as an informal assessment of each team member. Formal peer evaluations were conducted three to four times during the semester to confirm informal team member assessments provided in the weekly progress reports. |

Student evaluations provided one means of evaluating students' perceptions of the effectiveness of course structure and teaching methods. Evaluations included ratings of interpersonal skills, problem solving skills, and technology skills development. Student evaluation responses for six sections of the project course were summarized to assess achievement of course learning objectives. Student responses were based on a seven-point Likert scale ("Not at All" = 1, "A Great Deal" = 7). Responses were grouped to determine the percentage of students having unfavorable, neutral, or favorable ratings for each learning objective. Results are shown in Table III. These levels of learning outcome achievement surpassed the CIS department's goal of 80 percent favorable agreement on student surveys.

Table III. Achievement Measures for the Course Learning Objectives

| Learning Objective | Unfavorable Response Percentage | Neutral Response Percentage | Favorable Response Percentage | Average Response (7 pt. scale) | n |
|---|---|---|---|---|---|
| Team building | 2.0 | 5.0 | 93.1 | 6.34 | 101 |
| Technology skills | 5.0 | 5.0 | 90.0 | 6.21 | 100 |
| Problem analysis | 3.0 | 7.0 | 90.0 | 6.20 | 100 |
| Problem solving | 2.9 | 6.8 | 90.3 | 6.19 | 103 |
| Critical thinking | 2.9 | 6.9 | 90.2 | 6.14 | 102 |
| Oral communications | 4.0 | 5.9 | 90.1 | 6.12 | 101 |
| Written communications | 6.7 | 9.6 | 83.7 | 5.73 | 104 |

## STUDENT AND INSTRUCTOR RESPONSE TO THE LEARNING ENVIRONMENT

Student evaluations provided indicators of success, but an equally important measure came from students' informal comments and my observations as instructor. Also, in the dual role of instructor and project manager, I balanced instructional responsibilities with client expectations. The methods used to interact with students were different than those normally used in traditional courses; i.e. I served as mentor (instructional) and supervisor (project manager). Initially, I had the sense that I was relinquishing control of the classroom setting, and relying on self-paced student learning. Over several semesters, I discovered that the different learning environment resulted in a new set of student learning behaviors.

Students were placed in an environment that required critical thinking. Solving a problem for a business client, someone who needed an operational information system, made the course a departure from the familiar academic setting. Students discovered that the answers were not in a textbook, but they found that several good questions to resolve were presented there. Students realized memorizing facts so that they may be recalled on exams did not work well for ill-structured problems. Students learned to distinguish important facts, to view the problem and its solution from a systems perspective, and to apply systems concepts in creating a solution. The dynamics of face-to-face interaction with a business client was the first encounter of this kind for many students, and it was an invaluable learning experience. Their reasoning processes matured.

Students enhanced their team building skills during the semester-long experience. Two factors were important in developing team skills: (1) team monitoring held each member to a high level of accountability; and (2) the level of work forced team members to rely on one another; i.e., an individual student or subset of the students could not carry the team. Weekly progress reports

and periodic peer evaluations kept students on notice that their activities and contributions were under scrutiny. A private e-mail message from me, in the role of project manager, brought the under-performing student back in line. Similarly, public recognition of a student for making a valued contribution to the team effort encouraged team play. Fundamentally, submission of weekly progress reports ensured progress was made weekly. Teams that worked at regular intervals minimized the number and severity of team problems. Workload distribution has been more difficult to assess. Recently, weekly progress reports included a detailed Gantt chart showing team member responsibilities for the current week and the upcoming week. When workload appeared disproportionate, a comment to the students who took on too much, or too little, corrected the assignment of activities among team members. This level of monitoring minimized the effect of dominant students and sent the message that every student was expected to contribute in a meaningful way.

Students developed or refined important interpersonal skills. The growth occurred in oral communications, written communications, leadership and initiative, and decision-making. Oral and written communications skills were enhanced from repetition – nine oral presentations and five written reports were produced in fulfilling the project deliverables. Oral presentations were conducted so that team members rotated speaking duties. Development of leadership, initiative, and decision-making skills were the result of requiring students to define the contents and objectives of the deliverables. The process began the first day of class when expectations of the project proposal and plan were discussed. Initially, students struggled with the concept that system development progress would be measured at discrete points in time, but that they must determine what form and how much progress was reasonable at each milestone review. In resolving this question, team leaders emerged, students showed initiative in investigating technical capabilities of programming tools and prototyping methodologies, and group decision-making became focused. When students asked me what I expected of each deliverable, my response placed the decision-making responsibility with them – what is required to deliver the system? Once students understood that they were accountable for decisions, they made better choices and they took ownership in their decisions.

The exceptional students accepted the challenges and emerged as team leaders. They exhibited initiative in moving the team from a concept to a working solution. Conversely, the marginal students struggled to find their niches and many of them failed to take advantage of the opportunities for personal development. The middle tier of students achieved the largest level of personal growth, both learning what to do from the team leaders and what not to do from the marginal students. All students recognized what they would do differently on their next team-based development project.

Healthy competition served as a motivational factor that resulted in high-quality solutions. The competitive element was most evident during prototype demonstrations. For example, if one team created a user interface element that improved ease of use and drew praise from the client, the other teams imitated and expanded its design in subsequent prototypes. In the process, students learned technical details of web development and the importance of user interface design. The competitive urge to build superior solutions fostered deeper and broader learning.

Scope creep was a difficult reality for students to accept. Unlike programming courses, where all requirements were known before coding began, the system design specifications were never complete. Even when students realized that many of the omissions resulted from their substandard efforts in the systems analysis and design course, they preferred to find an easy-to-implement means of dealing with changing requirements. Deciding when to accommodate changes, and how to address their impacts within the project schedule was a painful lesson. Students learned what they should have done differently during analysis (the previous course) and project planning (an earlier phase of the project course).

Students expanded their technical knowledge. They developed a deeper understanding of systems analysis and design, especially model construction, systems specification, and project management. Also, they expanded their knowledge of web programming and implementation. Typically, the web application was based on a Microsoft SQL Server database tied to web forms using either Active Server Pages (ASP) or Macromedia's Cold Fusion. The use of these tools in the market place gave students an incentive to master them. Many students acquired the implementation-level knowledge through self-paced learning. This learning model was reasonable, as they completed programming and database courses previously. In this sense, the experience provided an example of life-long learning.

Prototyping helped students learn the finer points of modeling system design decisions. Modeling skills developed with practice, and several iterations were needed to develop accurate and robust models. The initial design models from the systems analysis and design course were not sufficient for implementation. During prototyping, students used these rough designs to build functional elements of the system. Upon reflection, students realized the errors and omissions in the original design specifications, and they revised the models to capture the design decisions made during prototyping. Although this form of modeling was more descriptive than prescriptive, it served as a feedback loop between a functional element of the system and its paper-based specification.

Student learning was most effective when the business clients represented small business. Typically, their problems involved common situations that students had experienced, or ones that they could comprehend easily. Conversely, when problems of large organizations were used, students spent a disproportionate amount of time learning the complexities of the problem domain. Small business problems provided ample opportunity for students to master system modeling and implementation, to refine project management skills, and to enhance interpersonal skills.

Students left the course understanding that technical skills were necessary, but that they were not sufficient. They appreciated the role of interpersonal skills and problem solving skills in their development as information systems professionals.

## CONCLUSION

Using a two-semester course sequence to analyze, design, and implement web-based applications has become one of the strongest elements of the CIS undergraduate program. Students valued the experience for its challenges and professional development opportunities. Alumni held the

experience in high regard as preparation for their first professional assignments. The project course provided opportunities to enhance interpersonal skills and problem solving skills. Design of the course required a departure from the traditional academic setting. Students responded to the new environment enthusiastically. Student evaluations and informal observations indicate success of the approach.

The keys to success have been selecting manageable problems from the small business sector, encouraging students to take advantage of the opportunities for professional growth, and conducting the course from the dual perspective of instructor and project manager.

## REFERENCES

1. Abi-Raad, M. (2000). Systems Analysis with Attitude!, ITiCSE 2000, Helsinki, Finland, July 2000, 57-60.
2. Davis, G.B., Gorgone, J.T., Couger, J.D., Feinstein, D.L., Longenecker, H.E. Jr., (1997). IS' 97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, ACM SIGMIS Database, Vol. 28(1):101-194.
3. Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., Longenecker, H.E. Jr., (2003). IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, ACM SIGMIS The Data Base for Advances in Information Systems, Vol. 34(1):1-52.
4. Grundy, J. (1997). A Comparative Analysis of Design Principles for Project-based IT Courses, ACSE '97, Melbourne, Australia, 170-177.
5. Noll, C.L., Wilkins, M. (2002). Critical Skills of IS Professionals: A Model for Curriculum Development, Journal of Information Technology Education, Vol. 1(3):143-154.
6. Radcliff, D. (2001). Hot Skills for a Cold Market, *ComputerWorld*, November 12, 2001, http://www.computerworld.com/careertopics/careers/story/0,10801,65450,00.html
7. Sweat, J. (2001). Reality Check for New Grads, *InformationWeek*, June 11, 2001, http://www.informationweek.com/841/grads.htm