Developing the Systems Project Course

ABSTRACT: The 1986 and 1990 DPMA model curricula both include a Systems Project course (CIS/86-8 and IS-8). This course places emphasis on systems theory and user requirement-based problem solving. This paper describes the development of the Systems Project course. Initially, it addresses the definition of the course, including the selection of projects, the creation of teams, development and tracking of a project plan, the analysis and design of the system, and the completion of the project. It then discusses the software tools needed and the major problems that are encountered in teaching the Systems Project course. If developed and taught well, the Systems Projects course may very well be the most important class any information systems student could take.

Albert L. Harris
Department of Decision Sciences
John A. Walker College of Business
Appalachian State University
Boone,North Carolina 28608
Phone: (704) 262-6180
E-Mail: harrisal@mm.appstate.edu

KEYWORDS: Information Systems Curriculum, DPMA 1990 Guidelines, Systems Project Course

INTRODUCTION

The 1986 DPMA model curriculum [1] identifies the Systems Development Project (CIS/86-8) as a capstone course in the IS curriculum. In this course, emphasis is placed on the development of a computer application using the life cycle approach. Ideally, the class forms into project teams to analyze a system and develop the design for a replacement automated system. The new system is to be implemented to the extent possible. The 1990 DPMA model curriculum provides for a similar course, Systems Project (IS-8), and places an emphasis on systems theory and user requirement-based problem solving [2]. The ACM model curriculum also recognizes the importance of a capstone course by including IS10, Information Systems Project [3], in its model curriculum.

The Systems Project course can be one of the best learning tools available to an Information Systems (IS) major, if it is structured appropriately. Specifically, it provides students with a real-world situation and allows them to analyze and develop an automated solution to a business need. Students interact with business users, many of whom have little or no computer literacy. They are taken from the controlled environment of the classroom and put in the every day environment of the user, including telephones, questions, and other distractions. There is no book solution to the analysis or design undertaken by each team. The Systems Project course can also address one of the competencies employers say is essential for an IS professional - oral and written communications [4, 5, 6].

This course can also be one of the most difficult to teach for the very same reasons that it is a good learning tool. Real users are unpredictable. Interaction skills in a "work" environment are new to most students. The environment is uncontrolled, causing great stress on students and, many times, the instructor. There is a short time span for the project, usually one semester. Finally, there is no single answer to the analysis and design problem. Developing and leading the Systems Project course is a difficult and time-consuming task for the instructor.

DEFINING THE SYSTEMS PROJECT COURSE

The Systems Project course is a capstone course meant to integrate all the course work taken by an IS major and to give graduating seniors an introduction to a real systems analysis and design project. It is usually designed as a one semester course (as outlined in the DPMA and ACM curricula), which provides severe time constraints on the project (like most projects). If the school has a core curriculum with writing or speaking components, one or both of the components might be included.

It is important that the entire system development life cycle (or as much as possible) be included in the course. To accomplish this, the course should be developed so students will:

1. Develop a Project Proposal (or use a Project Proposal developed by the users or system owner)
2. Divide up into teams
3. Develop a project plan for the project
4. Analyze the current system
5. Develop a design of the new system
6. Complete the project or develop a prototype and demonstrate it to the user.

Each of these areas will be discussed in more detail in the following paragraphs.

Develop A Project Proposal

The best learning environment for this course is a project with real users, outside the classroom environment. One approach to identifying projects is for the instructor to seek appropriate projects from companies that wish to participate and have the users or system owner develop the Project Proposal. Instructors can identify potential companies through IS or business Advisory Councils, industry contacts, or by sending

letters to the local Chamber of Commerce or other local business organizations. In this way, the instructor can identify potential projects that he/she believes are appropriate and acceptable.

Another approach, and the one used in my class, is to make each student responsible for identifying a potential project for the course and preparing and presenting a Project Proposal. This requires that the student have preliminary discussions with the user. At a minimum, the Project Proposal should include (a) a description of the setting, (b) a description of the need or statement of the problem, and (c) a description of the project and its scope. The Project Proposal may also include a short paragraph on why the student thinks the project would be good for the course. If this approach is used, each student should make a five minute presentation on his/her Project Proposal to the class. This provides each student with practice in making oral presentations and gives the class members an idea of the projects that could be performed during the semester. After all the proposals are presented, projects to be used during the semester are chosen. Not all proposals are true candidates for selection, due to such factors as distance from campus, complexity, unwillingness of the user to provide hardware needed to implement the system, etc. Selection of projects can be made by the instructor, the class, or some combination after non-appropriate proposals are removed from consideration. Since the team approach is used in the course, only about one in three of the project proposals will result in actual projects, simulating the development of project priorities and the project backlog in IS.

Examples of some of the projects that have been proposed and performed over the past few years include:

- Permit Tracking System for a small town. This system tracked all building permits, including permit submission, required approvals, inspections, and completion.
- Front Desk System for a 70 room motel. The system included reservations, check-in, bill accumulation, and check-out/billing.
- Inventory and Customer Tracking Systems for a small music store. Inventory included all serialized music instruments. The system tracked customers who had purchased instruments, repeat sales, trade-ins, and produced labels for promotional mailings.
- Inventory and Check-out System for the Outdoor Programs department in the university. All canoes, bicycles, tents, and camping gear were tracked, including who checked-out and checked-in which gear and the resulting charges.
- Customer Tracking System for a small retail specialty store. Purchase amounts per customer were tracked, promotional mailing lists provided, and a yearly store credit based on customer purchases was computed and maintained.

Each of these system was developed for the specific needs of the client and provided unique challenges for student projects.

Create Teams

The next step is to divide the class into teams. The ideal team size seems to be three, but teams of two to four are possible. Three person teams allow adequate distribution of tasks, are good (or sometimes bad for the odd person out) for conflict resolution, and provide minimum scheduling problems. The person proposing a project that is chosen is automatically assigned to the team that will perform the project. Selection of other team members can be made by the instructor, the students, or some combination. While it would be best for the students to chose their own teams, this approach many times does not result in the best teams for the learning environment. Teams should be formed based on complimentary technical and interpersonal skills and gender/racial mixtures.

Develop a Project Plan

Once the teams are established, each team should develop a Project Plan for the project. The plan should include tasks, milestones, and deliverables scheduled over the semester. A good learning device is to have the teams estimate total hours that will be spent on each task. Students tend to grossly under- or over-estimate the hours. During the semester, students should keep time sheets, tracking the time spent on each task. Every two to three weeks, each team should submit an updated project schedule, showing actual progress and actual time spent on each task, and a short, written progress report. This keeps the team members and the instructor informed concerning the progress of the project. At the end of the semester, students are able to compare their estimates with what it actually took to do the project. This almost always proves to be an excellent learning experience.

Analyze the Current System

The first major deliverable of the team should be a system profile which includes a statement of the problem, identification of users, an analysis of the strengths and weaknesses of the current system, and an identification of potential problems and opportunities. This is followed by a physical data flow diagram (DFD) of the current system. Creation of these outputs will entail multiple visits of the project team with the user. At least three levels (context, systems, and at least one detailed level) of DFDs should be developed. This requires that the team study the current system in detail and understand the current system. Students should be encouraged to review the DFDs with the instructor and the user as they are developed.

Once the DFDs of the current system are completed, the team should make a formal presentation to the class and receive feedback. Invariably, holes in the DFD are found or unanswered questions are uncovered during the class presentation. After the presentation, the team should make any changes needed to the DFDs and present them to the user for review and sign-off. Entity-relationship diagrams (or other data modeling techniques), logical DFDs (or other process modeling techniques) and other tools (e.g. network modeling, data dictionaries/repositories, etc.) may also be developed, depending on the needs of the project and the tools the instructor wishes to emphasize. An analysis package, consisting of the project proposal, system profile, logical (and possibly physical) DFDs, copies of all source documents, reports and forms, data models, process models, detailed descriptions of all data flows, data stores, and processes and any other techniques used to analyze the system, should be prepared. This represents a complete analysis package and can be used for user acceptance, designing the new system and grading.

Design the New System

In this phase of the project, each team should develop a logical design of the new system, concentrating on the processes and data flows. A physical design of the new system may also begin to take shape. At a minimum, students should design, on paper, the menu structure, all the databases, all source documents, and all input screens, output screens and reports not part of the prototype. The database design should also be performed in this phase. Many times the team must work with the user to define hardware and software requirements for the system.

Complete the Project or Develop and Demonstrate the Prototype

The last step in the project is to complete the project and deliver a useable system or develop and demonstrate a prototype of the new system. The prototype approach is used many times because there is usually not enough time in a semester to develop and implement an entire system. This approach also gives students experience in prototyping methodology. Students can use a CASE tool or a database package (e.g. dBase, Paradox, etc.) to develop the system or prototype.

The prototype should include all menu screens, the major data entry screens, and at least one inquiry screen and report. Any screen or report not included in the prototype, must be designed on paper and must be part of the final design document. Each team should present the system or prototype to the class as a dry run and receive feedback. If revisions are needed, they can be made before the system or prototype is delivered to the user. There should be a formal acceptance of the system or prototype and final design document by the user.

A final design document should be prepared and should include all designed or prototyped menus, input and output screens, source documents, reports, and all other prepared documentation. Two copies of the final design document should be required. One copy is given to the user, while the second copy is retained by the instructor for grading, use by the students in their job search, and use by instructors in other courses, especially systems analysis and design courses. Also due with the final design document is the final updated Project Schedule and a diskette with a copy of all programs written for the system or prototype.

SOFTWARE TOOLS

One of the requirements of this type of course is that the required software tools be available. These include:

- Project management
- Lower CASE, flow diagramming (or data/process modeling) tools
- Upper CASE or a database package with programming capability (e.g. dBase, FoxPro, Access, Paradox, etc).

Since project teams are required to develop a Project Plan, it would be best to have a project management package available (e.g. Microsoft Project, Harvard Project Manager, CA-SuperProject, etc.). The requirements for the project management package are that it 1) is available to students, 2) allows students to schedule tasks, milestones and deliverables, 3) allows an actual vs. schedule comparison, and 4) allows the accumulation and comparison of estimated and actual hours worked by task.

To perform the analysis, a lower CASE or flow diagramming (or data/process modeling) package is needed to develop the DFDs, data or process models. CASE tools are excellent for the analysis. Flow diagramming packages, such as FlowDraw, Easyflow, or FlowCharting, allow students to automate and easily modify and change their DFDs and data/process models. Given the number of times most teams will revise their work, this becomes an indispensable tool.

Finally, an upper CASE or database package is needed to develop the system or prototype. Because of user requirements, teams may use different packages. Upper CASE tools present a problem for most teams because the developed systems are, many times, not able to run on the user's computer. In most cases, the user must procure a license to run the developed software on their computer. The availability of the database package and cost of the package to the user become concerns for the project team.

PROBLEMS

Developers of a Systems Project should realize that there are several, fairly major, problems that must be dealt with before a good course can be developed. First, is the problem of finding suitable projects. Instructors can usually draw on Advisory Councils, industry contacts, or local business organizations. Students can usually draw on their work experience, the work experience of friends, or general observations to identify possible projects. Because of the limited time frame (one semester), the biggest problem is usually with the project scope. Each project selected for a team study must have an appropriate scope, not too small yet not too large. The best projects are ones that can be implemented on a microcomputer or an existing company minicomputer/mainframe. Users must be willing to spend time with the students, several hours during the first few weeks and then one to two hours a week for the remainder of the semester. Users must also either have the appropriate hardware (computer, communications links, bar code scanners, printers, etc.) and software (database package) or be willing to procure them. Users must realize that the project is a learning vehicle and that a completed system may not be available at the end of the semester. If the scope is appropriate and the users understand the needs and constraints, then the project is a candidate for the class.

Once the project is actually started, the time constraints of a semester make controlling the projects (i.e. keeping projects on schedule) the next major problem. Students and instructors find that one semester is usually not enough time to perform all the activities needed to complete most projects. Development of project proposals, selection of the projects to be performed and creation of the teams must be completed in a minimum of time, usually within seven to nine days after the start of the semester. The project plan is due two to seven days after the projects are selected and the teams formed. The initial meeting of the team with the user must be scheduled as soon as possible. One of the best ways to control the project is to ensure that a realistic project plan is developed. It is imperative that the instructor closely monitor each team's progress and not allow the teams to fall behind schedule. The analysis should be completed by the mid-term in the semester. Finally, enough time must be allowed for each team to develop the final design document and present their system or prototype to the class and the user for acceptance.

The third major problem is scheduling appropriate times when all team members can meet. All team members must attend the team meetings with the user. The instructor may sometimes allow team meetings to take place during the time class would normally meet, but this alone is not enough time for the teams to interact. Common available time should be considered when forming teams.

Team interaction is the fourth major problem. Students find that, for the first time, they have to interact with other students on a professional or work level. This means each student must assume responsibility for their part of the project. It is a time for exchange of ideas among the team members. It is also a time for conflict. Many times two team members will chose one solution or approach over the one proposed by the third team member. This causes considerable stress and tension among team members. Some major prob-

lems can develop among team members in this type of situation. The instructor must be aware of conflicts among team members and work with team members to resolve problems. One approach used when conflicts appear is to schedule a meeting with the team members and the instructor each week to review progress and resolve conflicts. This way the instructor can act as facilitator to the group.

From an instructor viewpoint, the number of students and the number of projects in the class are important. The instructor should attend three team-user meetings at a minimum. These should include the first meeting, the analysis review meeting and the system or prototype demonstration. Many times the instructor attends other meetings with each team to discuss progress and/or resolve conflicts. This adds to the scheduling of these user meetings. Teams should know that nothing is shown to the user before it has been reviewed by the instructor. The proper teaching of this course takes considerably more time from the instructor than most courses. Consequently, the ideal number of students per class is about 12 to 15. With five teams or fewer, sufficient time is available in class for project presentations and for significant instructor-team interaction.

While this discussion has not identified all the problems, it has focused on the major ones that have appeared over the years of teaching the Systems Projects course.

CONCLUSIONS

The Systems Project is quite possibly the most important class any IS major could take. It combines a real-world project with significant user-student and student-student interaction. If designed and implemented correctly, students are able to utilize the analysis and design skills learned in the IS curriculum and experience the system development life cycle in action. They also are given many opportunities to practice their oral and written communications skills. Finally, they are able to create an impressive analysis and design document that can help in their job search by demonstrating their analytical, technical and written communications skills. The Systems Project course should be a must in any IS curricula and should be used as the capstone course for any IS major.

REFERENCES

1. *CIS '86 The DPMA Model Curriculum for Undergraduate Computer Information Systems*, 2nd Edition, Data Processing Management Association, Park Ridge, Il. 1986

2. *Information Systems, The DPMA Model Curriculum for a Four-Year Undergraduate Degree*, Data Processing Management Association, Park Ridge, Il. 1990

3. *Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs*, Education Board of the Association of Computing Machinery, New York, NY, 1983

4. Nance, K. S., "The Effectiveness of Model Curricula in Addressing Skills Needed by Information Systems Students," *Interface*, 14(2), Summer, 1992, pp. 2-7

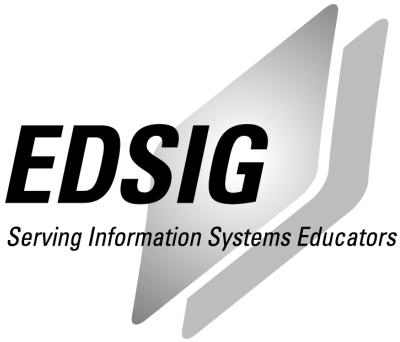
5. Pollack, T. H. "The MIS Curriculum: Which Competencies are Really Important to Business Professionals?" *Proceedings of the Eight Information Systems Education Conference*, Chicago, October 13-14, 1990, pp. 76-80

6. Sumner, M., Klepper, R, and Schultheis, R., "An Assessment of the Attitudes of Graduates and Employers Toward Competencies Needed for Entry-Level MIS Positions," *Proceedings of the Eight Information Systems Education Conference*, Chicago, October 13-14, 1990, pp. 129-134

Author's Biography

Albert L. Harris, PhD, CDP, CSP, CISA, CMC is an Associate Professor at the John A. Walker College of Business at Appalachian State University, Boone, NC. He received his Ph.D. in Management Information Systems (MIS) from Georgia State University. He has over 15 years of information systems consulting experience. He has served as Regional Director of Consulting for a national CPA firm and Training Manager for a large hospital information system software developer. He has taught the Systems Project equivalent course at Appalachian State University for the past five years. He has published articles in the International Journal of Management, the Journal of Computer Information Systems, the International Journal of Computer Applications in Technology, Information and Management, the Journal of Management Consulting, and Computerworld. Dr. Harris is a Certified Systems Professional (CSP), a Certified Information Systems Auditor (CISA), a Certified Data Processor (CDP), and a Certified Management Consultant (CMC). He is a member of the Decision Sciences Institute, the Institute of Management Sciences, the International Association for Computer Information Systems, and EDSIG.

Information Systems & Computing
Academic Professionals



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096