

USING 4GL TOOLS IN PROJECT-ORIENTED COURSES

Linda Salchenberger

Loyola University of Chicago
Department of Management Science
10 East Pearson
Chicago, Illinois 60611

Abstract: The widespread availability of fourth generation languages (4GL's) and CASE (computer-aided software engineering) tools has presented students with the opportunity to design and implement real-world systems as class projects within the span of a one-semester course. The use of these tools during the systems development life cycle is discussed based on experiences with a microcomputer-based system which was developed by students in an upper division MIS class. The methodology used for directing student projects and some particular problems faced when using 4GL's as development tools are addressed.

KEYWORDS: 4GL, Systems Development Life Cycles, CASE Tools

INTRODUCTION

The development of systems for class projects provides valuable hands-on experience for students and bridges the gap between understanding and applying structured analysis and design principles. Since MIS students are preparing to become systems analysts, it is worthwhile for them to experience some of the "real-world" problems they will encounter during the software development process while they still have the opportunity to solve them with the guidance and support of an instructor and a peer group.

Currently, many large and small businesses have purchased microcomputer hardware and software but have not really been involved in any formal systems development procedures. They use word-processing and spreadsheet software for stand-alone applications, but have not yet committed to developing integrated business systems. They are anxious to receive any help they can get to gain maximum benefits from their microcomputers. Therefore, there are many opportunities for student involvement in the design and implementation of business systems, benefitting both students and clients.

In this article, a case study is presented which demonstrates the successful application of structured software engineering principles to the development of a system using CASE and 4th generation development tools. Details of an application which was designed and implemented as a student project are given. The use of prototyping in all stages of the systems life cycle is discussed. Finally, some general rules of thumb for applying fourth generation technology to the systems development process are presented.

OVERVIEW OF THE CLASS PROJECT

Since there is no formal MIS project course in our curriculum, the project is integrated in the Management Information Systems course. The goals of the project are consistent with the goals of this course which focuses on management information systems in the business environment. The students enrolled in this course successfully completed a systems analysis and design course and had some programming experience with a 3GL (COBOL and/or PL/1). To prepare for work on the project,

we read a number of articles which discussed the successful and unsuccessful implementation of information systems, the use of prototyping for design and development of application systems, and the nature of user involvement in design and development. (A bibliography is available upon request).

Much has been written about the project-oriented approach in software engineering or systems analysis and design classes (1), (2), and (5). Several factors are repeatedly cited as critical factors for successful student projects. These include the level of complexity of the project, the background and preparation of the students and the instructor, and the development of and adherence to a project schedule. We addressed each of these issues before selecting a project. The proposed project seemed feasible because we were able to partition it into a set of tasks which could be accomplished within the allotted time. Students had programming experience and had used CASE tools in other courses and were thus prepared for work on the project. Finally, weekly meetings were set up with the client to avoid missing scheduled due dates.

PROJECT DESCRIPTION

The proposed system was a PC-based management information system to be developed for a major computer hardware and software retail store and installed on a local area network. The system would maintain a large inventory database, produce weekly management reports, and facilitate ad-hoc queries. A customer tracking system would be used by sales to target groups for promotions. The customer data base would also be used by the service department to maintain repair and maintenance information. Other subsystems included payroll and performance evaluation and a prospect tracking system for identifying certain target groups for special promotions. Once fully operational, the system would be installed on the local area network consisting of 8 workstations and one dedicated server within the store. The system would eventually provide remote access for the outside sales force using laptop computers over phone lines.

The main work to be completed by the students was the overall design of the system and the implementation of the inventory subsystem. The inventory module was developed first and served as a prototype to demonstrate the report and query features to the store personnel. This key module was selected because the data it maintained was required by other subsystems. Other modules would be completed at a later date by the store software support staff, using the inventory module as a model.

The development tool selected was a relational database (Paradox) which includes an application generator and a programming language (PAL). The choice was made by the store manager, after determining that the students felt the tool would be appropriate for the intended application. The students were not familiar with Paradox and some learning time was built into the project schedule.

PROJECT SCHEDULE

The first step of the project was to complete a schedule which included the major milestones and their proposed

completion dates. Deliverables were identified as an initial analysis document, the design of an integrated system, the implementation of a key module of the system, and a final report which would include system documentation. The students submitted a proposal which detailed these items and their due dates to the instructor and the potential client. Many modifications were made to the schedule as time went on, but every effort was made to complete the deliverables promised in the original proposal.

there are many opportunities for student involvement in the design and implementation of business systems, benefitting both students and clients.

Successful completion of the project in the allotted time required that many phases of the project be conducted in parallel. There were several front-end issues which could not be allowed to delay the project and some decisions were delayed while other work continued. For example, we were unable to determine exactly how inventory data would be downloaded from the mainframe; management said they would make the decision at a later date. This did not prevent us from going ahead with the database design under the assumption that we would be able to access the data in a specific format. Weekly meetings with the store manager requiring progress

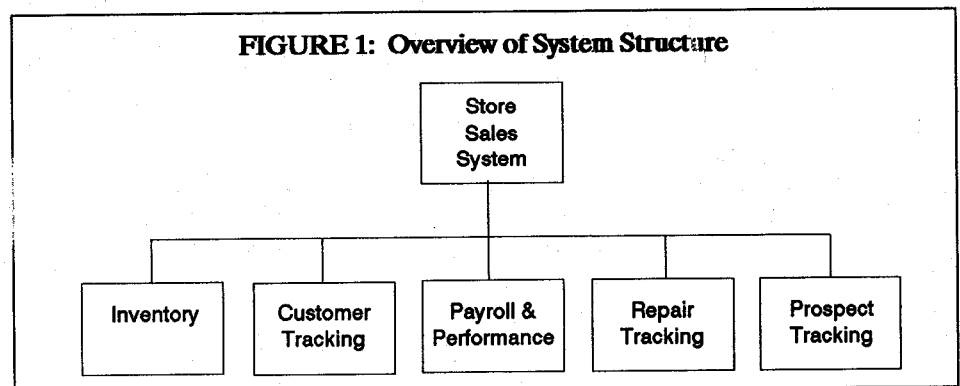
reports provided an incentive to the students to complete the tasks assigned.

REQUIREMENTS ANALYSIS

The first major task was to develop an informal requirements analysis which was documented in narrative form. The store manager and sales staff reviewed and revised the initial version. After acceptance of the preliminary analysis, a system prototype was developed for user review. (Prototyping was used during each stage of the project and proved to be an excellent communication tool between the students and the client.) The use of Paradox's application generator facilitated the development of the prototype which proved to be useful in requirements analysis. The translation of the requirements from a document to an actual working prototype provided the user with a more realistic model of the proposed system. As a result, he evaluated the requirements in a more detailed fashion and many more changes were suggested as a result of seeing the prototype in operation.

ANALYSIS AND DESIGN

A structured analysis and design approach was used and Excelerator, a CASE tool, was used to complete context diagrams for the system and detailed data flow diagrams for the inventory module. The main modules were identified as the inventory subsystem, the customer (current and prospective) tracking subsystems, the repair subsystem, and sales performance/commission subsystem. (See Figure 1).



Data flow diagrams were used to document the design and proved to be an effective communication tool as well as an excellent analysis and design tool. Detailed data flow diagrams and a data dictionary became part of the system documentation. Excelerator was used to integrate the data dictionary and data flow diagrams and thus, complete the analysis and design of the proposed system. The system design was presented to the endusers and their valuable input was used to make any necessary changes before the actual programming was begun.

DATA DESIGN

The design of the database was based on existing mainframe files (source of data for reports) and user report requirements. While prototyping facilitated the design effort, overall design priorities differed slightly from those usually adopted with rapid prototyping (3). Since a relational database package was used to implement the system, the format of reports and screens could be easily changed by the user after the system was in place. Therefore, it was not necessary to specify the format of the output and input screens and reports in great detail.

The majority of the effort was spent on the overall database analysis and design, including the identification of the objects of interest, the related data, the relationships between the objects, and the tables needed to define the relationships and support the data entry functions. Four tables were developed including an inventory master table, "tickets" or invoice table, customer master table, and inventory transaction table. Two more tables were required to facilitate the data entry process. These were a tickets-customer table which supported a screen used to enter invoice and customer data and a transaction table which defined a one-to-many relationship between invoice and transaction details. These tables were normalized by the students.

Users found the schema easy to understand and seemed comfortable with the relational data base framework. The

real power of the relational data base software was demonstrated by its ability to define subschemas for different applications while maintaining the integrity of the database. If we had been restricted to a 3GL development environment, this type of application would have been impossible to design and develop within our time constraints.

DEVELOPMENT, TESTING, AND IMPLEMENTATION

Two crucial features of the systems development strategy were the use of structured techniques and prototyping. A topdown approach was used; in some cases, sections of code were developed by different student groups. Prototyping was consistently used throughout this phase of the life cycle to present the user with working versions of the system. Since live data had not yet been entered into the system, test data was developed and used for testing. Training was limited to working with the store manager who had been involved in the project from its inception. Plans for user training were left with the manager, but time prohibited any other student involvement in that activity. Every effort was made to "self-document" the system by using variable names which were representative of the attribute and providing comments within the programs written by the students. The design documents and a final report completed the documentation process.

A post-implementation review was conducted informally, during a phone conversation with the store manager. Errors and omissions in the system were largely due to the students' ignorance of business procedures. The system was successfully implemented and is currently operational, because of the manager's continued interest and a staff member's assistance.

PROBLEMS

Not unlike many microcomputer-based applications, one of the major problems was the access to existing data. The inventory tracking system required that the current inventory data which was

on a mainframe in a nonstandard format be downloaded into the pc-based system. Weeks were spent attempting to determine how this would be accomplished. Management was supportive, but as usual, the information systems staff was overworked and not willing to assist with this problem. Some calculations revealed that with an on-line data entry form, staff at the store could manually enter the data in about three weeks time. This meant that special data entry screens had to be designed, but this was not a difficult or time-consuming task using the integrated relational data base software which had been selected for the project. This still left the problem of weekly updates to the system which would be done manually until another solution could be found.

If we had been restricted to a 3GL development environment, this type of application would have been impossible to design and develop within our time constraints.

Many lessons were learned when students began to design the logic of the update system. We found that some programming paradigms which are trivial to code using a 3GL are actually very difficult to implement using the database programming language. At one point, we were unable to code a required algorithm and received assistance from the software vendor who admitted to the limitations of the programming capability of the software. This experience showed that the power of a 4GL tool does not come without a cost.

One of the weaknesses of relational databases is their inefficiency when large data sets are involved, and this case was no exception. The problem was resolved by adding memory to the 386-based machine to which workstations were networked and using disk-caching techniques. Plans to continue to upgrade the hardware were in the final report.

Maintaining a high level of individual student involvement can be problematic in class projects. In the past, I have used a team approach with different teams for analysis, design, development, and implementation. This was unsatisfactory because it was difficult to maintain the same level of involvement in the project for each of the teams. During past projects, lack of communication between the groups resulted in inconsistencies, errors and omissions. A slightly different project structure was used in directing this project and proved to be more satisfactory. Although different groups were responsible for the analysis, design, and implementation, the entire class was involved in the actual programming effort. The inventory subsystem was divided into three parts and each group of 3 or 4 students generated a section of code. This meant all the students become proficient in Paradox and there was greater involvement in analysis and design when each group presented their work. The small class size was an important factor in successfully implementing this concept.

CONCLUSIONS

Several lessons learned while working on this project can be applied to directing student projects which utilize CASE tools and 4GL's. It is likely that students will not be experienced in using the 4GL development tool which businesses will prefer since most institutions teach programming with 3GL's like COBOL and PASCAL. If the students have the dual problem of developing the system and learning how to use the particular software, a good strategy is to assign program modules to be developed during the learning process which can later be used in the system. Most small business systems require the development of familiar programming paradigms and after they are developed in key components of the system, they can often be used in many different parts of the system. Good documentation must be developed during the design phase since there is no time to document the system after it is developed. It is especially important to leave the client with this

documentation since the student is not a permanent employee and cannot be consulted at a later date.

The best student evaluations from the MIS course over the past 5 years are from the years when a real-world project was completed, rather than a case ("toy project") was developed.

The role of the user during the design and development process is rapidly changing (4). CASE tools and techniques allow the user to participate more actively in the design of report and screen prototypes and free the developer for more technical tasks. We were especially fortunate to have a sophisticated user group who were willing to learn and experiment and provided excellent feedback during the requirements analysis and design phases. A crucial part of our design strategy was to implement what researchers have told us for years; namely that iterative designs driven by user feedback are important for the success of a system.

CASE tools like Excelerator can be used to implement structured analysis and design techniques. If a CASE tool is not available, any graphics tool which has the capability of drawing data flow diagrams will be helpful. Student projects should focus on the application of structured methods to systems development rather than detailed design of outputs. With 4GL's, it is relatively easy for users to modify the format of their reports at a later date. The greatest benefit can be realized if the student consultants develop the logic of essential functions of the system. While users are willing to spend some time learning the query language of a system, they should be spared the burden of attempting to write programs to perform routine processes.

The traditional systems development life cycle can be modified so that several aspects of a project are worked on concurrently ("fast-tracking"). The development of the prototype with user feedback is facilitated by a 4GL. The use of a 4GL tool for systems development requires a development strategy which may differ from traditional strategies because of the nature of the tool itself. The nature of systems development is changing, with more emphasis on specifying requirements accurately and less on programming, as more tools become available to support that part of the development effort.

While directing student projects requires preparation, organization, and commitment from the project leader (the classroom instructor), nevertheless, it is effective as a pedagogical method, based on student feedback. The best student evaluations from the MIS course over the past 5 years are from the years when a real-world project was completed, rather than a case ("toy project") was developed. It is difficult to simulate the development of information systems in the business environment; perhaps one needs to experience it.

Key steps to success include investigating business contacts, screening project proposals for feasibility, maintaining a project schedule, and involving all the students in the development process. A client must be found who is truly interested and has the time to spend with students; in some cases, potential clients may be found within the university. Finally, the use of 4GL tools and prototyping makes the rapid systems development needed when time is limited possible.

REFERENCES

1. Carver, Doris. "Software Engineering for Undergraduates." ACM SIGCSE Bulletin 16, No. 3. (1984): 23-26.
2. Chrisman, C. and B. Beccue. "Updating Systems Development Courses to Incorporate Fourth Generation Tools", ACM SIGCSE Bulletin 17, No. 1. (1985): 109-113.

3. Jenkins, A. Milton. "Prototyping: A Methodology for the Design and Development of Application Systems." *Spectrum* 2, No. 2. (1985).
4. Rosson, Michael, Stephen Maass, and William Kellogg. "The Designer as User: Building Requirements for Design Tools from Design Practice", *Communications of the ACM* 31, No. 11, (1988): 1288-1299.
5. Tharp, Alan. "The Impact of Fourth Generation Programming Languages." *ACM SIGCSE Bulletin* 16, No. 2. (1984): 37-41.

AUTHOR'S BIOGRAPHY

Linda Salchenberger is an Assistant Professor of Management Science at Loyola University of Chicago, Chicago, Illinois. Her research areas of interest include structured analysis and design and expert systems. She has consulting experience in developing decision support systems for small businesses.



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.

Copyright ©1991 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096