IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4840

Student Approaches to Projects: Software Engineering vs. Information Systems

Anthony Scime
Department of Computer Science
State University of New York College at Brockport
350 New Campus Drive, Brockport, NY 14420, USA
phone: (585) 395-2323 fax: (585) 395-2304
ascime@brockport.edu

ABSTRACT

Software engineering and information systems analysis and design work toward the same result. They are derived from different origins; yet use the same tools and techniques. They may use different approaches to achieve the same results. This paper discusses teams of students from these disciplines working together to build a system.

INTRODUCTION

There are many similarities between software engineering and information systems analysis and design. Software engineering and information systems both involve the design, development, implementation, support and management of software artifacts (Bagert et al., 1999; Davis et al., 1997; Information Technology Association of America, 1997). These software artifacts may be program modules, methods to store and retrieve data, or representations of either.

These similarities exist within the theory of software engineering and information systems. In practice, the approach to artifact creation is different. As a practical demonstration of the differences a group of information systems students completed a requirements analysis and Unified Modeling Language (UML) design as a class project. Then two teams completed the project's software construction and implementation. One implementation team consisted of two information systems students, who had been involved in the design. The other team had two computer science students with strong software engineering backgrounds.

Despite using the same theoretical symbol set and inquiry methods, the different origins of software engineering and information systems caused the students to apply different methods. This paper discusses the analysis, design, construction, and implementation of a work scheduling and billing information system for health care aides in an assisted living facility. It illustrates the differences between information systems and software engineering in solving implementation problems.

THEORETICAL SIMILARITY

Software work can be classified into four common categories of work: conceptualizing, developing, modifying, and supporting. Conceptualizers are involved with the creation of ideas concerning the basic nature of a computer system artifact. Software conceptualizers investigate new ways of processing, storing, transmitting, and representing information. Developers are people who specify, design, construct, and test an artifact. They apply existing technology to new problems. Modifiers maintain systems by making improvements to increase the efficiency of information processing, storage, or communication. Supporters provide system administration and help users in operating computer systems (Freeman & Aspray, 1999). Either software engineering or information systems professionals may fill these jobs.

Both disciplines use the same symbolic systems to define the language for communication. The symbols used for discourse revolve around mathematical and graphical modeling, programming languages, digital logic, and assembly languages. Both disciplines also use the same inquiry methods to answer or solve problems. These methods define the collection, organization, and analysis of requirements and design of software systems (Scime, 2002).

The native problem solving approach in information systems and computer science/software engineering are the same. Computing professionals typically solve problems both adaptively and innovatively. An adaptive style is one that works within the accepted scope of a problem not changing basic assumptions or design. Innovators are more likely to attempt a novel solution. Computing problem solving styles are equally balanced between adaptation and innovation (Sim & Wright, 2002).

Although much is common, the origins of software engineering and information systems are different. Professional fields are the applied application of knowledge developed in theoretical disciplines (Stark & Lattuca, 1997). The founding disciplines of software engineering are mathematics and engineering; and of information systems - management and mathematics (Scime, 2002).

THE PROJECT

Hill Top Apartments is a 150 unit senior living, church affiliated, nonsectarian, not-for-profit corporation. The facility, located in a medium size US city, consists of one and two bedroom apartments. It is a self-contained living facility that boasts fine dining, in-house entertainment, a hair salon, a heated indoor swimming pool, post office, bank and other amenities needed for the daily needs of the residents.

As a separate service, care and companion assistance is provided to residents. This assistance ranges from escorts to the in-house facilities through housekeeping, cooking, laundering and even dressing and bathing. Living assistance is also provided to senior citizens that live in their homes throughout the city. This includes exceptional services such as snow plowing, lawn care, and shopping. The client, whether in-house or home-care, is charged for the aide's time based on the service provided.

The services are grouped together and called programs. Programs are the basis for the rates charged. In Shared Care one aide provides the service to two or three clients simultaneously. One-on-One care is when the aide's time is totally devoted to one client. Community Care applies when an aide is directing a group activity; the cost of the aide is divided among the participating clients. Contract Care is used primarily in the home-care situation where exceptional services are performed.

Fifty mostly part-time home health aides provide these personal care services 24 hours a day to approximately one hundred clients. Depending on the service, an aide may spend all-day or only 15-minutes with a client. Because of the variety of services, scheduling problems were common place. Clients that required service for which aides were not scheduled led to client dissatisfaction. Unrecorded services led to

missed billing opportunities. Management realized automating the scheduling and billing process could relieve these problems.

The Process

The client requests assistance from the scheduling clerk in the assisted living office (Figure 1). This assistance may be a one-time activity or a reoccurring activity, such as assistance with getting out of bed and dressing daily. The scheduler coordinates all the clients' activities and schedules the aides. A daily activity schedule is produced for each aide. This activity schedule may contain ad-hoc tasks, routine tasks, or a combination of both. Ad-hoc tasks are those that only occur once or very infrequently. Routine tasks are activities performed daily. Most of the tasks are routine, so an aide follows a predefined route during their shift. The aide annotates any changes on the schedule as they occur. At the end of the aide's shift the annotated schedule is returned to the scheduler and the schedule is up-dated to reflect the activities as they occurred.

Completed and corrected schedule information (client, activity, duration) is passed to the billing system. The billing system produces a monthly invoice for each client specifying the activity, date, duration, and cost. This invoice is passed to the accounting department, which bills the client.

Hill Top requested assistance in improving the assisted living services operation through automation of the scheduling and billing of services. The goals of this project were to:

- Automate the scheduling process.
- Based on the scheduling information generate billing invoices.
- Have the capability to generate reports.

The project was divided into to phases, each one semester long. During the first phase a team of 9 information systems students analyzed the current processes and designed an information system. The second phase consisted of a team of 4 students to construct and implement the designed system.

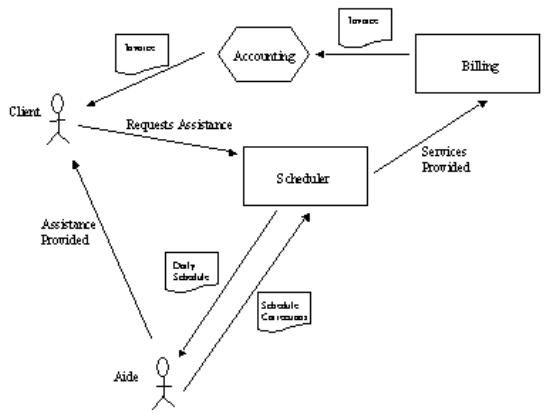
Requirements Analysis and Design

The requirements analysis and design phase resulted in a Unified Modeling Language (UML) model (use cases, class diagrams, and sequence diagrams) of the proposed system with user interface screens designed in detail. This was documented in a design specification and presented to the Assisted Living Manager.

Early on in the analysis it was determined that the management of Hill Top was not prepared to re-engineer the entire scheduling and billing system. A solution was needed that would just assist the scheduling and the billing clerks in completing their tasks. Management clearly indicated they did not want an integrated solution. The project was divided into sub-projects one for scheduling and one for billing. The work of the two groups was coordinated to provide for future integration of the sub-systems, and to provide for a common 'look and feel' between the sub-systems.

Over the course of a month, the two teams met separately with the manager. The aim of these meeting was to learn the processes by which the Assisted Living Office performed the scheduling and billing tasks, evaluate the current systems in place, and review available documentation. The teams presented their understanding of the requirements and an outline of proposed scheduling and billing solutions to the Assisted Living Manager. After receiving approval from the manager, detailed designs were prepared. The two teams spent approximately two months designing the new scheduling and billing systems. Periodic reviews were held with the manager to evaluate progress and compliance with the requirements. At the end of the Spring semester, the team presented the complete design for the two systems. The designs called for an MS-Access database with tables, canned queries, user interface screens, and pre-defined reports.

Figure 1. The Aide Scheduling and Billing Process



System Construction and Implementation

At the beginning of the second phase (the following Fall semester) two teams were again formed to construct and implement the designs. Because of graduations and other commitments, of the 9 students who completed the analysis and design only 2 students were available for construction and implementation. Both of these students had worked on the scheduling sub-system analysis and design. These two information systems students continued with the scheduling sub-system of the project.

A second set of 2 students was recruited to build and implement the billing portion from the already completed design. These two students were computer science/software engineering students knowledgeable in UML, but not experienced with databases or MS-Access.

The teams were asked to provide a schedule for the construction and implementation through project completion. When the two teams came together to discuss their plans there were clearly differences in approach.

The scheduling team planned to follow an information systems approach. They began by constructing the database tables and defining the relationships between the tables. They then created the queries, the forms, and the reports in successive developments.

The billing team took a different approach. Their schedule followed a black-box programming method, developed as a result of their programming background. The black-box method calls for the development and testing of separate parts of the system. With this method the billing team developed a module from each use case, associated classes, and sequence diagrams in the project design. These modules were developed and tested independently. After all the modules were completed they were linked together by a main menu.

Documentation of the system was completed by the creation of user manuals and system manuals. A template for how the manuals were to be formatted was first created to insure uniformity between the scheduling and billing portions of the system.

Scheduling Sub-System

The implementation of the scheduling software began with a review of the requirements and design documents. The scheduling team began system implementation by building the designed tables for the database in accordance with the design specification.

The data entry forms were constructed. All of the forms were easy to create from the design, with one exception. The Scheduling Form presented some problems because of the complexity of the SQL statement needed to populate the form.

The scheduling form is a key form in the scheduling sub-system. It is the building block for the reports requested by the client. This form is used to schedule an aide and a client together for a particular service time. On the Scheduling Form the clerk would select a client that required service within a predefined time frame (i.e. a week or day). The form's underlying queries would then find an aide that was available to fill this time slot, and check to ensure that this aide had not already been scheduled. The underlying queries of the form required accessing mul-

tiple tables and proved difficult in MS-Access. It became necessary for the team to learn Visual Basic, as a program was needed to populate the form. It also required re-design of some of the database tables to make the data accessible.

Testing was completed on approximately three weeks worth of data to insure that the system was working properly. No apparent errors in the system were found. Finally, four reports that were required by the client were created, as well as, the menu system (switchboard) interface to complete the sub-system.

Billing Sub-System

One week into the construction and implementation phase the billing team noticed a discrepancy involving the number of different programs that a client of Hill Top could be a member. According to the Assisted Living Manager, and the design, a client can belong to more than one Hill Top program at a time. However, the design permitted multiple forms on which manipulating the client's program was possible. This allowed the service program attribute value to be changed during processing. The students recognized this as a typical synchronization update problem that could compromise data integrity and create multiple miss-billings of clients. Some re-design was necessary.

To solve this design problem the billing team modified the forms to move the location for program selection. The billing clerk would choose a client's programs only from the Charges Form.

A second problem was the billing reports. The reports use case and related classes were completely left out of the design specification. After consultation with the Assisted Living Manager on reporting requirements, that portion was designed and implemented. Because of the black-box approach, work on the other modules continued while these problems were being addressed.

A final billing team concern was their inexperience with MS-Access. Initially this was thought a problem, but the solid programming background of the team quickly compensated and the team gained the necessary skills.

CONCLUSION

Although coming from different backgrounds and using different approaches, the students produced similar and compatible information systems. The scheduling team's familiarity with the project was only a small advantage. The scheduling (information systems) team used a database approach. First the database tables were constructed and populated with test data. Then the queries necessary for the forms and reports were built. Finally the system was tested.

The billing (software engineering) team's knowledge of UML allowed them to understand the design and build the desired system. They used the use cases, associated classes, and sequence diagrams extensively to black-box program the required modules. After all the modules were completed and tested they were combined into the system.

Technically the resulting system was a great success. All 11 students (9 in analysis and design and 4 in construction and implementation, with 2 in both.) learned a great deal about systems requirements elicitation, system design, software and database construction, and systems implementation. They learned the importance of good design documentation. They also learned about the difficulties in dealing with a real client unfamiliar with computers and automation. The resulting system was designed to improve on the existing system, and was fundamentally built to the design. Clearly it was a technical and educational success.

There is a shortage of skilled software professionals to fill the needs of industry (Information Technology Association of America, 2001). Graduates of schools that have programs for conceptualizers, developers and modifiers are well prepared to fill positions in software development and database development. It matters little whether the education is in Software Engineering, Information Systems, or Computer Science.

REFERENCES

Bagert, D. J., Hilburn, T. B., Hislop, G., Lutz, M., McCracken, M., & Mangal, S. (1999). Guidelines for Software Engineering Education Version 1.0. [Technical Report CMU/SEI-99-TR-032] Software Engineering Institute, Pittsburgh, PA: Carnegie Mellon University.

Davis, G. B., Gorgone, J. T., Couger, J. D., Fienstein D. L., & Longnecker, H. E. (1997). IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Association of Information Technology Professionals.

Freeman, P., & Aspray, W. (1999). The Supply of Information Technology Workers in the United States. Washington DC: Computing Research Association.

Information Technology Association of America (ITAA) (1997). Help Wanted: The Workforce Gap at the Dawn of a New Century. Arlington, VA: Author.

Information Technology Association of America (ITAA) (2001). When Can You Start? Building Better Information Technology Skills and Careers. Arlington, VA: Author.

Sim, Edward R. & Wright, George (2002). A Comparison of Adaption-Innovation Styles Between Information Systems Majors and Computer Science Majors, Journal of Information Systems Education, 13(1), 29-35.

Scime, Anthony (2002). Information Technology Model Curricula Analysis, chapter 12 in Challenges of Information Technology Education in the 21st Century, E. Cohen, ed., Hershey, PA., Idea Group Publishing: 222-239.

Stark, J. S., & Lattuca, L. R. (1997). Shaping the College Curriculum: Academic Plans in Action. Needham Heights, MA: Allyn and Bacon, pp.113 – 178.