

A CASE STUDY OF THE APPLICATION OF THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC) IN 21ST CENTURY HEALTH CARE: SOMETHING OLD, SOMETHING NEW?

Mark McMurtrey

University of Central Arkansas

MarkMc@uca.edu

ABSTRACT

The systems development life cycle (SDLC), while undergoing numerous changes to its name and related components over the years, has remained a steadfast and reliable approach to software development. Although there is some debate as to the appropriate number of steps, and the naming conventions thereof, nonetheless it is a tried-and-true methodology that has withstood the test of time. This paper discusses the application of the SDLC in a 21st century health care environment. Specifically, it was utilized for the procurement of a software package designed particularly for the Home Health component of a regional hospital care facility. We found that the methodology is still as useful today as it ever was. By following the stages of the SDLC, an effective software product was identified, selected, and implemented in a real-world environment. Lessons learned from the project, and implications for practice, research, and pedagogy, are offered. Insights from this study can be applied as a pedagogical tool in a variety of classroom environments and curricula including, but not limited to, the systems analysis and design course as well as the core information systems (IS) class. It can also be used as a case study in an upper-division or graduate course describing the implementation of the SDLC in practice.

Keywords

Information systems, information systems curriculum, teaching information systems, systems development life cycle (SDLC), systems analysis and design, case study.

INTRODUCTION

The systems development life cycle, in its variant forms, remains one of the oldest and yet still widely used methods of software development and acquisition methods in the information technology (IT) arena. While it has evolved over the years in response to ever-changing scenarios and paradigm shifts pertaining to the building or acquiring of software, its central tenants are as applicable today as they ever were. Life-cycle stages have gone through iterations of different names and number of steps, but at the core the SDLC is resilient in its tried-and-true deployment in business, industry, and government. In fact, the SDLC has been called one of the two dominant systems development methodologies today, along with prototyping (Piccoli, 2012). Thus, learning about the SDLC remains important to the students of today as well as tomorrow.

This paper describes the use of the SDLC in a real-world health care setting involving a principle component of a regional hospital care facility. The paper can be used as a pedagogical tool in a systems analysis and design course, or in an upper-division or graduate course as a case study of the implementation of the SDLC in practice. First, a review of the SDLC is provided, followed by a description of the case study environment. Next, the application of the methodology is described in detail. Following, inferences and observations from the project are presented, along with lessons learned. Finally, the paper concludes with implications for the three areas of research, practice, and pedagogy, as well as suggestions for future research.

BACKGROUND

The SDLC has been a part of the IT community since the inception of the modern digital computer. A course in Systems Analysis and Design is requisite in most Management Information Systems programs (Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior, and de Vreede, 2010). While such classes offer an overview of many different means of developing or acquiring software (e.g., prototyping, extreme programming, rapid application development (RAD), joint application development (JAD), etc.), at their heart such programs still devote a considerable amount of time to the SDLC, as they should. As this paper will show, following the steps and stages of the methodology is still a valid method of insuring the successful deployment of software. While the SDLC, and systems analysis and design in general, has evolved over the years, at its heart it remains a robust methodology for developing software and systems.

Early treatises of the SDLC promoted the rigorous delineation of necessary steps to follow for any kind of software project. The Waterfall Model (Boehm, 1976) is one of the most well-known forms. In this classic representation, the methodology involves seven sequential steps: 1) System Requirements and Validation; 2) Software Requirements and Validation; 3) Preliminary Design and Validation; 4) Detailed Design and Validation; 5) Code, Debug, Deployment, and Test; 6) Test, Preoperations, Validation Test; and 7) Operations, Maintenance, Revalidation. In the original description of the Boehm-Waterfall software engineering methodology, there is an interactive backstep between each stage. Thus the Boehm-Waterfall is a combination of a sequential methodology with an interactive backstep (Burbach, 2004).

Other early works were patterned after the Waterfall Model, with varying numbers of steps and not-markedly-different names for each stage. For example, Gore and Stubbe (1983) advocated a four-step approach consisting of the study phase, the design phase, the development phase, and the operation phase (p. 25). Martin and McClure (1988) described it as a multistep process consisting of five basic sequential phases: analysis, design, code, test, and maintain (p. 18). Another widely used text (Whitten, Bentley, and Ho, 1986) during the 1980s advocated an eight-step method. Beginning with 1) Survey the Situation, it was followed by 2) Study Current System; 3) Determine User Requirements; 4) Evaluate Alternative Solutions; 5) Design New System; 6) Select New Computer Equipment and Software; 7) Construct New System; and 8) Deliver New System.

Almost two decades later, a book by the same set of authors in general (Whitten, Bentley, and Dittman, 2004) also advocated an eight step series of phases, although the names of the stages changed somewhat (albeit not significantly). The methodology proceeded through the steps of Scope definition, Problem analysis, Requirements analysis, Logical design, Decision analysis, Physical design and integration, Construction and testing, and ending with Installation and delivery (p. 89). It is interesting to note that nearly 20 years later, the naming conventions used in the newer text are almost synonymous with those in the older work. The Whitten and Bentley (2008) text, in its present form, still breaks up the process into eight stages. While there is no consensus in the naming (or number) of stages (e.g., many systems analysis and design textbooks advocate their own nomenclature (c.f. Whitten, Bentley, and Barlow (1994), O'Brien (1993), Taggart and Silbey (1986)), McMurtrey (1997) reviewed the various forms of the life cycle in his dissertation work and came up with a generic SDLC involving the phases of Analysis, Design, Coding, Testing, Implementation, and Maintenance.

Even one of the most current and popular systems analysis and design textbooks (Kendall and Kendall, 2011) does not depart from tradition, emphasizing that the SDLC is still primarily comprised of seven phases. Although not immune to criticism, Hoffer, George, and Valacich (2011) believe that the view of systems analysis and design taking place in a cycle continues to be pervasive and true (p. 24). Thus, while the SDLC has evolved over the years under the guise of different combinations of naming conventions and numbers of steps or stages, it remains true to

form as a well-tested methodology for software development and acquisition. We now turn our attention to how it was utilized in a present-day health care setting.

Case Study Setting

The present investigation regards the selection of a software package by a medium-size regional hospital for use in the Home Health segment of their organization. The hospital (to be referred to in this monograph by a fictitious name, General Hospital) is located in the central portion of a southern state in the USA, within 30 minutes of the state capital. Its constituents reside in the largest SMSA (standard metropolitan statistical area) in the state and consist of both rural, suburban, and city residents. The 149-bed facility is a state-of-the-art institution, as 91% of their 23 quality measures are better than the national average (“Where to Find Care”, 2010). Services offered include Emergency Department, Hospice, Intensive Care Unit (ICU), Obstetrics, Open Heart Surgery, and Pediatrics. Additional components of General Hospital consist of an Imaging Center, a Rehabilitation Hospital, Four Primary Care Clinics, a Health and Fitness Center (one of the largest in the nation with more than 70,000 square feet and 7,000 members), a Wound Healing Center, regional Therapy Centers, and Home Care (the focal point of this study).

There are more than 120 physicians on the active medical staff, over 1,400 employees and in excess of 100 volunteers (“General Hospital”, 2010). In short, it is representative of many similar patient care facilities around the nation and the world. As such, it provides a rich environment for the investigation of using the SDLC in a 21st century health care institution.

Home Health and Study Overview

Home Health, or Home Care, is the portion of health care that is carried out at the patient’s home or residence. It is a participatory arrangement that eliminates the need for constant trips to the hospital for routine procedures. For example, patients take their own blood pressure (or heart rate, glucose level, etc.) using a device hooked up near their bed at home. The results are transmitted to the hospital (or in this case, the Home Health facility near General Hospital) electronically and are immediately processed, inspected, and monitored by attending staff.

In addition, there is a Lifeline feature available to elderly or other homebound individuals. The unit includes a button worn on a necklace or bracelet that the patient can push should they need assistance (“Home Health”, 2010). Periodically, clinicians (e.g., nurses, physical therapists, etc.) will visit the patient in their home to monitor their progress and perform routine inspections and maintenance on the technology.

The author was approached by his neighbor, a retired accounting faculty member who is a volunteer at General Hospital. He had been asked by hospital administration to investigate the acquisition, and eventual purchase, of software to facilitate and help coordinate the Home Health care portion of their business. After an initial meeting to offer help and familiarize ourselves with the task at hand, we met with staff (i.e., both management and the end-users) at the Home Health facility to begin our research.

THE SDLC IN ACTION

The author, having taught the SAD course many times, recognized from the outset that this particular project would indeed follow the stages of the traditional SDLC. While we would not be responsible for some of the steps (e.g., testing, and training of staff), we would follow many of the others in a lockstep fashion, thus, the task was an adaptation of the SDLC (i.e., a software acquisition project) as opposed to a software development project involving all the stages. For students, it is important to see that they benefit from understanding that the core ideas of the SDLC can be adapted to fit a “buy” (rather than “make”) situation. Their knowledge of the SDLC can be applied to a non-development context. The systematic approach is adaptable, which makes the knowledge more valuable. In this project, we used a modified version of the SDLC that

corresponds to the form advocated by McMurtrey (1997). Consequently, we proceed in this monograph in the same fashion that the project was presented to us: step by step in line with the SDLC.

Analysis

Problem Definition

The first step in the Systems Development Life Cycle is the Problem Definition component of the Analysis phase. One would be hard-pressed to offer a solution to a problem that was not fully defined. The Home Health portion of General Hospital had been reorganized as a separate, subsidiary unit located near the main hospital in its own standalone facility. Furthermore, the software they were using was at least seven years old and could simply not keep up with all the changes in billing practices and Medicare requirements and payments. The current system was not scalable to the growing needs and transformation within the environment. Thus, in addition to specific desirable criteria of the chosen software (described in the following section), our explicit purpose in helping General was twofold: 1) to modernize their operations with current technology; and 2) to provide the best patient care available to their clients in the Home Health arena.

A precursor to the Analysis stage, often mentioned in textbooks (e.g., Valacich, George, and Hoffer, 2009) and of great importance in a practical setting, is the Feasibility Study. This preface to the beginning of the Analysis phase is oftentimes broken down into three areas of feasibility:

- Technical (Do we have the necessary resources and infrastructure to support the software if it is acquired?)
- Economic (Do we have the financial resources to pay for it, including support and maintenance?)
- Operational (Do we have properly trained individuals who can operate and use the software?).

Fortunately, these questions had all been answered in the affirmative before we joined the project. The Director of Information Technology at General Hospital budgeted \$250,000 for procurement (thus meeting the criteria for economic feasibility); General's IT infrastructure was more than adequate and up to date with regard to supporting the new software (technical feasibility); and support staff and potential end users were well trained and enthusiastic about adopting the new technology (operational feasibility). Given that the Feasibility Study portion of the SDLC was complete, we endeavored forthwith into the project details.

Requirements Analysis

In the Requirements Analysis portion of the Analysis stage, great care is taken to ensure that the proposed system meets the objectives put forth by management. To that end, we met with the various stakeholders (i.e., the Director of the Home Care facility and potential end-users) to map out the requirements needed from the new system. Copious notes were taken at these meetings, and a conscientious effort to synthesize our recollections was done. Afterwards, the requirements were collated into a spreadsheet for ease of inspection (Exhibit 1). Several key requirements are described here:

MEDITECH Compatible: This was the first, and one of the most important requirements, at least from a technological viewpoint. MEDITECH (Medical Information Technology, Inc.) has been a leading software vendor in the health care informatics industry for 40 years ("About Meditech", 2009). It is the flagship product used at General Hospital and is described as the number one health care vendor in the United States with approximately 25% market share ("International News", 2006). All Meditech platforms are certified EMR/EHR systems ("Meditech News", 2012). "With an Electronic Health Record, a patient's record follows her electronically. From the

physician's office, to the hospital, to her home-based care, and to any other place she receives health services, and she and her doctors can access all of this information and communicate with a smartphone or computer” (“The New Meditech”, 2012). Because of its strategic importance to General, and its overall large footprint in the entire infrastructure and day-to-day operations, it was imperative that the new software would be Meditech-compatible.

Point of Care Documentation: Electronic medical record (EMR) point-of-care (POC) documentation in patients' rooms is a recent shift in technology use in hospitals (Duffy, Kharasch, Morris, and Du, 2010). POC documentation reduces inefficiencies, decreases the probability of errors, promotes information transfer, and encourages the caregiver to be at the bedside or, in the case of home care, on the receiving end of the transmission.

OASIS Analyzer: OASIS is a system developed by the Centers for Medicare & Medicaid Services (CMS), formerly an agency of the U.S. Department of Health and Human Services, as part of the required home care assessment for reimbursing health care providers. OASIS combines 20 data elements to measure case-mix across 3 domains—clinical severity, functional status and utilization factors (“Medical Dictionary”, 2010). This module allows staff to work more intelligently, allowing them to easily analyze outcomes data in an effort to move toward improved clinical and financial results (“Butte Home Health”, 2009). Given its strategic link to Medicare and Medicaid reimbursement, OASIS Analyzer was a “must have” feature of the new software.

Physician Portal: The chosen software package must have an entryway for the attending, resident, or primary caregiver physician to interact with the system in a seamless fashion. Such a gateway will facilitate efficient patient care by enabling the physician to have immediate access to critical patient data and history.

Other “Must Haves” of the New Software: Special billing and accounts receivable modules tailored to Home Health; real-time reports and built-in digital dashboards to provide business intelligence (e.g., OASIS Analyzer); schedule optimization; and last, but certainly not least, the system must be user friendly.

Desirable, But Not Absolutely Necessary Features: Security (advanced, beyond the normal user identification and password type); trial period available (i.e., could General try it out for a limited time before fully committing to the contract?).

Other Items of interest During the Analysis Phase: Several other issues were important in this phase:

- Is the proposed solution a Home Health-only product, or is it part of a larger, perhaps enterprise-wide system?
- Are there other modules available (e.g., financial, clinical, hospice; applications to synchronize the system with a PDA (Personal Digital Assistant) or smart phone)?
- Is there a web demo available to view online; or, even better, is there an opportunity to participate in a live, hands-on demonstration of the software under real or simulated conditions?

We also made note of other observations that might be helpful in selecting final candidates to be considered for site visits. To gain insight into the experience, dependability, and professionalism of the vendors, we also kept track of information such as: experience (i.e., number of years in business); number of clients or customers; revenues; and helpfulness (return e-mails and/or phone calls within a timely manner or at all).

Finally, some anecdotal evidence was gathered to help us evaluate each vendor as a potential finalist. For instance, Vendor A had an Implementation/Installation Team to assist with that stage of the software deployment; they also maintained a Knowledge Base (database) of Use Cases/List Cases describing the most frequently occurring problems or pitfalls. Vendor C sponsored an annual User Conference where users could share experiences with using the product, as well as provide feedback to be incorporated into future releases. To that end, Vendor C also had a user representative on their Product Advisory Board. Vendor E offered a “cloud computing” choice, in that the product was hosted in their data center. (A potential buyer did not have to choose the web-enabled solution.) Vendor E’s offering was part of an enterprise solution, and could be synchronized with a PDA or smart phone.

Design

As previously noted, for this particular case study of software selection, the researchers did not have to proceed through each step of the SDLC since the software products already existed. Thus, the Design stage of the SDLC has already been carried out by the vendors. In a similar vein, the coding, testing, and debugging of program modules had too been performed by each vendor candidate. Thus, after painstakingly analyzing all the wares, features, pros and cons, and costs and benefits associated with each product, we were now ready to make a choice: we would whittle our list of five potential vendors down to the two that we felt met our needs and showed the most interest and promise.

The Choice

The principle investigators arranged another meeting with the primary stakeholders of General Hospital’s Home Health division. After all, although we had done the research, they were the ones that would be using the system for the foreseeable future. As such, it only made sense that they be heavily involved. This is in line with what is put forth in systems analysis and design textbooks: user involvement is a key component to system success. Having carefully reviewed our research notes, in addition to the various brochures, websites, proposals, communications, and related documents from each of our shortlist of five vendors, together as a group we made our decision. We would invite Vendor B for a site visit and demonstration.

Vendor B was very professional, courteous, prompt, and conscientious during their visit. One thing that greatly supported their case was that their primary business model focused on Home Health software. It was, and still is, their core competency. In contrast, one other vendor (not on our original short list of five) came and made a very polished presentation, in the words of the Director. However, this company was a multi-billion dollar concern, of which Home Health software was only a small part. Thus the choice was made to go with Vendor B.

Ironically, this seller’s product was not Meditech compatible, which was one of the most important criteria for selection. However, through the use of a middleware company that had considerable experience in designing interfaces to be used in a Meditech environment, a suitable arrangement was made and a customized solution was developed and put into use. The middleware vendor had done business with General before and, therefore, was familiar with their needs.

Implementation

As is taught in SAD classes, the implementation stage of the SDLC usually follows one of four main forms. These are, according to Valacich, George, and Hoffer (2009): 1) Direct Installation (sometimes also referred to as Direct Cutover, Abrupt, or Cold Turkey method) where the old system is simply removed and replaced with the new software, perhaps over the weekend; 2) Parallel Installation, when the old and new systems are run side-by-side until at some point (the “go live” date) use of the former software is eliminated; 3) Single Location Installation (or the

Pilot approach) involves using one site (or several sites if the software rollout is to be nationwide or international involving hundreds of locations) as beta or test installations to identify any bugs or usage problems before committing to the new software on a large scale; and 4) Phased Installation, which is the process of integrating segments of program modules into stages of implementation, ensuring that each block works before the whole software product is implemented in its entirety.

The Home Care unit of General Hospital utilized the Parallel Installation method for approximately 60 days before the “go live” date. Clinicians would “double enter” patient records and admissions data into both the old and new systems to ensure that the new database was populated, while at the same time maintaining patient care with the former product until its disposal. The Director of the Home Care facility noted that this process took longer than anticipated but was well worth it in the long run. Once the “go live” date was reached the new system performed quite well, with a minimal amount of disruption.

Training of staff commenced two weeks before the “go live” date. Of the approximately 25 users, half were trained the first week and the rest the next. Clinicians had to perform a live visit with one of their patients using the new system. Thus they would already have experience with it in a hands-on environment before switching to the new product and committing to it on a full-time basis.

It is again worth noting that the implementation method, Parallel Installation, follows from the SDLC and is what is taught in modern-day SAD courses. Thus, it was satisfying to the researchers that textbook concepts were being utilized in “real world” situations. It also reinforced that teaching the SDLC was in line with current curriculum guidelines and should continue.

Maintenance/Support

Software upgrades (called “code loads” by the vendor) are performed every six weeks. The Director reported that these advancements were not disruptive to everyday operations. Such upgrades are especially important in the health care industry, as changes to Medicare and billing practices are common occurrences. The Director also noted that all end users, including nurses, physical therapists, physicians, and other staff, were very happy with the new system and, collectively, had no major complaints about it. General Hospital expects to use the software for the foreseeable future, with no plans to have to embark on another project of this magnitude for quite some time.

DISCUSSION

Many inferences and observations were gleaned by both the researchers and hospital staff during the course of the investigation. First, we all learned that we must “do our homework”; that is, much research and analysis had to be performed to get up to speed on the project. For instance, while the principle investigators both had doctoral degrees in business administration, and one of them (the author) had taught the systems analysis and design course for over ten years at two different institutions, neither of us had any practical experience in the Home Health arena. Thus, we had to familiarize ourselves with the current environment as well as grasp an understanding of the criteria set forth by the stakeholders (both end-users and management). This was an important lesson learned, because we teach our students (in the SAD class) that they must not only familiarize themselves with the application at hand, but they must also interact with the users. Much research has been conducted in the area of user involvement and its relationship to system success (e.g., Ives and Olson, 1984; Baroudi, Olson, and Ives, 1986; Tait and Vessey, 1988). Therefore it was satisfying, from a pedagogical standpoint, to know that concepts taught in a classroom setting were being utilized in a real-world environment.

It was also very enlightening, from the standpoint of business school professors, to see how the core functional areas of study (e.g., marketing, management, accounting, etc., not to mention MIS) were also highly integral to the project at hand. During our research on the various vendor companies, we were subjected to a myriad of different marketing campaigns and promotional brochures, which typically touted their wares as the “best” on the market. Key, integral components (such as billing, scheduling, business intelligence, patient care, electronic medical records (EMR), etc.) that are critical success factors in almost any business were promoted and we were made keenly aware of their strategic importance. Again, this was very rewarding from the point of view from business school professors: we were very pleased that our graduates and students are learning all of these concepts (and more) as core competencies in the curriculum.

Finally, probably the most positive outcome from the project was that patient care will be improved as a result of this endeavor. Following that, it was enlightening that an adaptation of the SDLC was applied to a healthcare setting and it achieved positive results. This showed that the SDLC, in part or in whole, is alive and well and is an important part of the MIS world in both practice and academia. In addition, key outcomes regarding each were identified and are elaborated upon in the following section.

IMPLICATIONS FOR PRACTICE, RESEARCH AND PEDAGOGY

Implications for Practice

This project, and case study, was an application of pedagogy on a real-world systems analysis project. As such, it has implications for practice. First, it showed that concepts learned in a classroom environment (such as the SDLC in the systems analysis and design course) can be effectively applied in a business (or in our case, a health care) environment. It was very satisfying for us, as business school professors, to see instructional topics successfully employed to solve a real-world problem. For practitioners, such as any organization looking to acquire a software package, we hope that we have shown that if one applies due diligence to their research effort that positive outcomes can be achieved. Our findings might also help practitioners appreciate that tried and true methods, such as the SDLC, are applicable to projects of a similar nature, and not just academic exercises to fulfill curriculum requirements. We find this among the most gratifying implications.

Implications for Research

This project could be used as the beginning of a longitudinal study into the life cycle of the Home Health software product selected. It is customary to note that maintenance can consume half of the IS budget when it comes to software, especially large-scale systems (Dorfman and Thayer, 1997). It would be interesting to track this project, in real time, to see if that is indeed the case. Furthermore, an often-neglected phase of the SDLC is the stage at the very end: disposal of the system. By following the present study to the end, it would be enlightening (from all three viewpoints of research, practice, and pedagogy) to see what happens at the end of the software’s useful life. Additional future research might investigate the utilization of the SDLC in different contexts, or even other settings with the healthcare arena.

Implications for Pedagogy

Insights for the SAD Course

After learning so much about real-world software acquisition throughout this voluntary consulting project, the author has utilized it in classroom settings. First, the obvious connection with the SAD course was made. To that end, in addition to another semester-long project they work on in a group setting, the students pick an application domain (such as a veterinary clinic, a dentist’s office, a movie rental store, etc.) and perform a research effort not unlike the one described in this monograph. Afterwards, a presentation is made to the class whereby three to five

candidate vendors are shown, along with the associated criteria used, and then one is chosen. Reasons are given for the selection and additional questions are asked, if necessary. This exercise gives the students a real-world look at application software through the lens of the SDLC.

While some SAD professors are able to engage local businesses to provide more of a “real-world” application by allowing students to literally develop a system, such an endeavor was not possible at the time of this study. The benefits of such an approach are, of course, that it provides students “real world” experience and applying concepts learned in school to practical uses. The drawback is that it requires a substantial commitment from the business and oftentimes the proprietors pull back from the project if they get too busy with other things. Thus, the decision was made to allow students to pick an application domain, under the assumption that they had been contracted by the owners to acquire a system for them.

Such an exercise enables students to engage in what Houghton and Ruth (2010) call “deep learning”. They note that such an approach is much more appropriate when the learning material presented involves going beyond simple facts and into what lies below the surface (p. 91). Indeed, this particular exercise for the SAD students was not rote memorization of facts at a surface level; it forced them to perform critical thinking and analysis at a much greater depth of understanding. Although the students were not able to complete a “real world” project to the extent that other educators have reported (e.g., Grant, Malloy, Murphy, Foreman, and Robinson (2010), the experience did allow students to tackle a contemporary project and simulate the solving of it with real-world solutions. This gave them a much greater appreciation for the task of procuring software than just reading about it in textbooks. The educational benefits of using real-world projects are well established both in the United States (Grant et al., 2010) and internationally (Magboo and Magboo, 2003).

From an IS curriculum standpoint, this form of exercise by SAD students helps bridge the well-known gap between theory and practice (Andriole, 2006). As was shown in this monograph, the SDLC is a theory that has widespread application in practice. The project performed by students in the SAD class reinforces what Parker, LeRouge, and Trimmer (2005) described in their paper on alternative instructional strategies in an IS curriculum. That is, SAD is a core component of an education in information systems, and there is a plethora of different ways to deliver a rich experience, including the one described here.

Insights for IS Courses, SAD and non-SAD

Other insights gained, by the SAD students as well as the core MIS course, have to do with what the author teaches during the requisite chapter on software. In class, I present this topic as “the software dilemma”. This description is tantamount to the recognition that when acquiring software, businesses must make one of three choices (in general). The options are “make” versus “buy” versus “outsource” when it comes to acquiring software. (There is also a hybrid approach that involves customizing purchased software.)

Briefly explained, the “make” option presupposes that the organization has an IT staff that can do their own, custom, programming. The “buy” alternative relates to what was described in this paper, in that General Hospital did not have the resources to devote to developing software for their Home Health segment, and as such enlisted the researchers to assist in that endeavor. The “outsource” choice alludes to several different options available, under this umbrella, on the modern-day IT landscape. The decision to outsource could range from an application service provider (ASP) delivering the solution over the internet (or the “cloud”) to complete transfer of the IT operation to a hosting provider or even a server co-location vendor.

Thus, a project like this one could be used in the core MIS course to further illustrate problems and potential pitfalls faced by businesses, small and large, when it comes to software acquisition. Instructors could use the features of this case study to focus on whatever portion of it they

thought best: project management, budgeting, personnel requirements, marketing, etc. It could even be used in a marketing class to investigate the ways in which vendors, offering similar solutions to standard problems, differentiate themselves through various marketing channels and strategies.

Furthermore, the case study is ripe for discussion pertaining to a plethora of business school topics, from economics and accounting to customer relationship management. The case is especially rich fodder for the MIS curriculum: not only systems analysis and design, but programming and database classes can find useful, practical, real-world issues surrounding this case that can be used as “teaching tools” to the students.

Finally, a case study like this one could even be used in an operations management, or project management, setting. The discovery of issues, such as those raised in this paper, could be fruitful research for both undergraduate and graduate students alike. A team project, along with a group presentation as the finale, would also give students much-needed experience in public speaking and would help prepare them for the boardrooms of tomorrow.

CONCLUSION

Two business school professors, one an MIS scholar and the other retired from the accounting faculty, were called upon by a local hospital to assist with the procurement of software for the Home Health area. These academics were up to the challenge, and pleasantly assisted the hospital in their quest. While both researchers hold terminal degrees, each learned quite a bit from the application of principles taught in the classroom (e.g., the SDLC) to the complexities surrounding real-world utilization of them. Great insights were gained, in a variety of areas, and have since been shown as relevant to future practitioners (i.e., students) in the business world. It is hoped that others, in both academe and commerce, will benefit from the results and salient observations from this study.

REFERENCES

1. About Meditech (2009) Retrieved on May 19, 2010 from <http://www.meditech.com/AboutMeditech/homepage.htm>
2. Andriole, S. (2006) Business Technology Education in the Early 21st Century: The Ongoing Quest for Relevance. *Journal of Information Technology Education*, 5, 1-12.
3. Baroudi, J., Olson, M., and Ives, B. (1986, March) An Empirical Study of the Impact of User Involvement on System Usage and Information Satisfaction. *Communications of the ACM*, 29, 3, 232-238.
<http://dx.doi.org/10.1145/5666.5669>
4. Boehm, B. W. (1976, December) Software Engineering. *IEEE Transactions on Computers*, C-25, 1226-1241.
<http://dx.doi.org/10.1109/TC.1976.1674590>
5. Burbach, R. L. (2004) The Boehm-Waterfall Methodology, Retrieved May 20, 2010 from <http://infolab.stanford.edu/~burbach/watersluice/node52.html>
6. Butte Home Health & Hospice Works Smarter with CareVoyant Healthcare Intelligence (2009) Retrieved May 21, 2010 from http://www.carevoyant.com/cs_butte.html?fn=c_cs
7. Dorfman, M. and Thayer, R. M. (eds.) (1997) Software Engineering, IEEE Computer Society Press, Los Alamitos, CA.
8. Duffy, W. J. RN, Morris, J., CNOR; Kharasch, M. S. MD, FACEP; Du, H. MS (2010, January/March) Point of Care Documentation Impact on the Nurse-Patient Interaction. *Nursing Administration Quarterly*, 34, 1, E1-E10.

9. "General Hospital" (2010) *Conway Regional Health System*. Retrieved on May 18, 2010 from <http://www.conwayregional.org/body.cfm?id=9>
10. Gore, M. and Stubbe, J. (1983) *Elements of Systems Analysis* 3rd Edition, Wm. C. Brown Company Publishers, Dubuque, IA.
11. Grant, D. M., Malloy, A. D., Murphy, M. C., Foreman, J., and Robinson, R. A. (2010) Real World Project: Integrating the Classroom, External Business Partnerships and Professional Organizations. *Journal of Information Technology Education*, 9, IIP 168-196.
12. Hoffer, J. A., George, J. F. and Valacich, J. S. (2011) *Modern Systems Analysis and Design*, Prentice Hall, Boston.
13. "Home Health" (2010) *Conway Regional Health System*. Retrieved on May 18, 2010 from <http://www.conwayregional.org/body.cfm?id=31>
14. Houghton, L. and Ruth, A. (2010) Making Information Systems Less Scrugged: Reflecting on the Processes of Change in Teaching and Learning. *Journal of Information Technology Education*, 9, IIP 91-102.
15. "International News" (2006) Retrieved on May 19, 2010 from <http://www.meditech.com/aboutmeditech/pages/newsinternational.htm>
16. Ives, B. and Olson, M. (1984, May) User Involvement and MIS Success: A Review of Research. *Management Science*, 30, 5, 586-603.
<http://dx.doi.org/10.1287/mnsc.30.5.586>
17. Kendall, K. and Kendall, J. E. (2011) *Systems Analysis and Design*, 8/E, Prentice Hall , Englewood Cliffs, NJ.
18. Magboo, S. A., and Magboo, V. P. C. (2003) Assignment of Real-World Projects: An Economical Method of Building Applications for a University and an Effective Way to Enhance Education of the Students. *Journal of Information Technology Education*, 2, 29-39.
19. Martin, J. and McClure, C. (1988) *Structured Techniques: The Basis for CASE* (Revised Edition), Englewood Cliffs, New Jersey, Prentice Hall.
20. McMurtrey, M. E. (1997) *Determinants of Job Satisfaction Among Systems Professionals: An Empirical Study of the Impact of CASE Tool Usage and Career Orientations*, Unpublished doctoral dissertation, Columbia, SC, University of South Carolina.
21. "Medical Dictionary" (2010) *The Free Dictionary*. Retrieved May 21, 2010 from <http://medical-dictionary.thefreedictionary.com/OASIS>
22. "Meditech News" (2012) Retrieved April 1, 2012 from <http://www.meditech.com/AboutMeditech/pages/newscertificationupdate0111.htm>
23. O'Brien, J. A. (1993) *Management Information Systems: A Managerial End User Perspective*, Irwin, Homewood, IL.
24. Parker, K. R., Larouge, C., and Trimmer, K. (2005) Alternative Instructional Strategies in an IS Curriculum. *Journal of Information Technology Education*, 4, 43-60.
25. Piccoli, G. (2012) *Information Systems for Managers: Text and Cases*, John Wiley & Sons, Inc., Hoboken, NJ.
26. Taggart, W. M. and Silbey, V. (1986) *Information Systems: People and Computers in Organizations*, Allyn and Bacon, Inc., Boston.

27. Tait, P. and Vessey, I. (1988) The Effect of User Involvement on System Success: A Contingency Approach. *MIS Quarterly*, 12, 1, 91-108.
<http://dx.doi.org/10.2307/248809>
28. "The New Meditech" (2012) Retrieved April 1, 2012 from
<http://www.meditech.com/newmeditech/homepage.htm>
29. Topi, H., Valacich, J., Wright, R., Kaiser, K., Nunamaker Jr, J., Sipior, J., and de Vreede, G.J. (2010) IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*, 26, 1, 1-88.
30. Valacich, J.S., George, J. F., and Hoffer, J. A. (2009) Essentials of System Analysis and Design 4th Ed., Prentice Hall , Upper Saddle River, NJ.
31. "Where to Find Care" (2010) Retrieved on May 18, 2010 from
<http://www.wheretofindcare.com/Hospitals/Arkansas-AR/CONWAY/040029/CONWAY-REGIONAL-MEDICAL-CENTER.aspx>
32. Whitten, J. L. and Bentley, L. D. (2008) Introduction to Systems Analysis and Design 1st Ed., McGraw-Hill, Boston.
33. Whitten, J. L., Bentley, L. D., and Barlow, V. M. (1994) Systems Analysis and Design Methods 3rd Ed. Richard D. Irwin, Inc., Burr Ridge, IL.
34. Whitten, J. L., Bentley, L. D., and Dittman, K. C. (2004) Systems Analysis and Design Methods 6th Ed., McGraw Hill Irwin, Boston.
35. Whitten, J. L., Bentley, L. D., and Ho, T. I. M. (1986) Systems Analysis and Design, Times Mirror/Mosby College Publishing, St. Louis.