Capstone Experiences: Lessons Learned in Mastering CS Competencies

Farid Hallouche, Scott James, John Hansen, Moe Bidgoli

***Abstract*— It is the goal of every undergraduate Computer Science/Information Systems program to educate and train highly skilled graduates who are well prepared to enter the specialized workforce. Much time and debate is spent on developing the appropriate curriculum to aid in this necessary preparation. As we have learned, however, curriculum alone is not enough to provide the complete answer. Indeed, it clearly appears that real world capstone experiences that combine theory and practice are required to hone an undergraduate's skill set prior to graduation. This paper will examine the various facets of our system that we employ with a view to allowing our students to acquire a set of competencies that are highly in demand in the current markets as they evolve.**

***Index Terms*— Undergraduate Research/Capstones, Theory, Practice, Skills, Computer Science Education.**

I. Introduction

Our department is home to two-degree programs: Computer Science and Computer Information Systems. Both programs have some common classes such as Programming I and II (in C++) and Windows Programming (using Visual Basic). In addition, there are other courses that are similar, however the Computer Science program tends to be much more theoretic in nature, while the Computer Information Systems program leans towards applied concepts. Many Computer Science majors earn a minor in mathematics, whereas the Computer Information Systems majors typically earn minors in business. A student tends to major in one program or the other often based on how mathematically inclined that student is.

It has been the tradition of the Computer Information Systems program to offer capstone experiences since the program's inception in 1986. Only recently has the Computer Science program begun to offer capstone experiences for students who choose the software engineering or networking cognates within the program.

It is through the authors' leading work, that several upper division classes have been retooled to provide students with highly meaningful classroom experiences, which will prepare them for their capstone projects in either program. In particular, we have found that there are three fundamental changes that need to be made to the curricula: (1) both theory and practice must be taught. Many traditional programs would concentrate on one or the other almost exclusively; (2) in addition to the use of technology, well equipped laboratories will need to be set up in order to help students carry out hands-on experiments and programming assignments, and (3) offer real world capstone case studies through projects with entities outside of the university. We will now present some of our experiences and results in each of these areas in detail.

Manuscript received July 30, 2008.

Farid Hallouche is with the Computer Science and Information Systems Department, Saginaw Valley State University, 7400 Bay Rd, University Center, MI 48710, USA (Tel.: 989-964-2799; Fax: 989-964-2796; e-mail: hallouch@svsu.edu).

Scott James is with the Computer Science and Information Systems Department, Saginaw Valley State University, 7400 Bay Rd, University Center, MI 48710, USA (e-mail: james@svsu.edu).

John Hansen is with the Computer Science and Information Systems Department, Saginaw Valley State University, 7400 Bay Rd, University Center, MI 48710, USA (e-mail: jhansen@svsu.edu).

Moe Bidgoli is with the Computer Science and Information Systems Department, Saginaw Valley State University, 7400 Bay Rd, University Center, MI 48710, USA (e-mail: bidgoli@svsu.edu).

II. Teaching Theory and Practice

As previously stated, many traditional Computer Science programs tend to be mathematically rigorous and theoretic with one goal being preparing students to continue on to graduate school. Unfortunately, a large number of students plan to make their bachelor's degree a terminal one. As it is, if a Computer Science program focuses more on theory, many students are woefully prepared to begin working for a potential employer in industry or business.

Likewise, many Computer Information Systems programs tend to offer more business courses than programming courses. As a result, students accumulate a fair amount of knowledge about attempting to manage the software development process, and are well versed in using end user applications such as word processors and spreadsheets. Unfortunately, these students have very little, if any, theory, and consequently have a good deal of trouble in developing application systems and programming. By contrast, the CIS students in our program, although still lacking in theory, spend much more time programming than they do in taking management classes.

To tackle this dilemma, a well-thought solution was adopted, that incorporates both theory and practice in the upper division classes, leading to the capstone experience classes. This solution seem to have generated so much enthusiasm amongst faculty and students alike, and has indeed created so many fabulous opportunities for skill development. We will cite two examples; one CIS and the other CS.

1) CIS Example

The first example we would like to cite comes from the 300-level Data Communications and Networking course that our CIS students are required to take. This class could be delivered completely as a lecture class in which the elements of Data Communications are taught. Typical coverage would look at such things as transmission media, modems, protocols and LAN software and hardware. In other words, some practical concepts are covered, but no underlying related theory whatsoever. To correct this problem, the practical knowledge is provided along with a little theory covering digital signals, the OSI and TCP/IP models and error correction mechanisms. The basic algebra and statistics involved are kept to a minimum so that are not anywhere as rigorous as the math in a Computer Science networking course.

In order to add more practice to the course, the students are required to use a programming language to implement various application systems that handle the SMTP, POP3, Finger, Time,

HTTP, FTP and RSH protocols. In addition, the students are required to build their own Cat5E cables with cutters, strippers and crimping tools to put RJ-45 terminators on the ends of the cable. The students then use wire-testing equipment to verify that they have built their patch cable correctly and that the cable works. There is a high degree of satisfaction from the students in the fact that they learned about the 4 pairs of twisted wire, and used that knowledge to actually build their own cables. These cables were then used subsequently to make their peer-to-peer network. This single course hands-on experience acquired helps our students to understand the subject material as well as gain a useful skill set. On the other hand, our CIS students will have received more theory than is traditionally taught in a CIS networking course with a business concentration.

2) CS Example

The second example of teaching theory and practice is related to a 400-level CS network operating system course. The aim of this course in combining theory and practice was from the very outset to help our students acquire 'academic knowledge' and 'industry knowledge', and to encourage active learning through interaction and creativity. The theory part, which is delivered mostly through lectures, covers the essentials of operating system architecture and design. This includes kernels, processes, threads, scheduling, memory management and multiprocessing. The practice part is delivered mainly through supervised hands-on lab sessions as well as programming assignments. The aim of this part is, on the one hand, to reinforce the theoretical concepts and help students understand the material content, on the other hand, to develop useful related skills.

In both the theory and the practice parts, case studies are provided that relate to Linux and Windows 2000 operating systems. The hands-on Labs and programming assignments cover multithreading and thread synchronization, process management, inter-process communication, Linux administration, security and programming, Windows 2000 administration, security and programming, active directory programming and security and .NET remoting. As in the case of the CIS example above, the experience of incorporating a balance of theory and practice has been very worthwhile. In particular, it has allowed for the teaching of a diverse content that was carefully distributed across the lecture and lab sessions.

III. SETUP OF WELL-EQUIPPED, DEDICATED LABORATORIES

Our department is very fortunate in that, in addition to the computing open Labs, we have setup two labs dedicated to network computing and hardware courses. We try to incorporate technology in as many classes as possible. We provide multiple development platforms and languages, and in many cases students are told that they may use whatever tools they find appropriate. Currently our labs offer the following software packages: Microsoft Visual Studio .NET, Java, DreamWeaver Studio MX, Microsoft SQL Server 2000, GNU C++, PHP, Apache, MySQL, Perl, Microsoft Exchange, Microsoft Systems Management Server, Linux, Solaris, Windows XP, Windows 2000 and XP. Students are encouraged to learn many different packages, and as they get into their upper division classes, to pick the appropriate packages based on the problem they are trying to solve.

The laboratory networks connect Pentium PCs and notebooks. Network devices and cabling were chosen to allow copper, fiber as well as wireless connectivity. One Windows 2000 server and a Linux server were installed and configured. These are operated from a single keyboard and monitor set via a KDM switch device, adequately ventilated and housed in a locked box for physical security. Wall-mounted racks are used that house such network devices as routers, switches and hubs. Shared network printers are connected directly to the network through their built-in Network Interface Cards (NIC). Some of the hardware devices used in our network implementation include CAT5 copper and fiber optic cables; Ethernet, fiber and wireless network adapters; Ethernet and fiber hubs; Ethernet and fiber switches; wireless access points; network testers and troubleshooters and a Cisco router.

IV. CAPSTONE EXPERIENCES

This involves a two-class capstone project. The goal is to enable students to develop skills in solving a large-scale project for a partner in industry. The concerned capstone faculty advisor locates potential industry partners with whom he then identifies projects that are appropriate in difficulty and duration as to provide a real-world experience for the students. Typical industry partners include not-for-profit organization or small commercial companies that cannot afford to have an I.T. staff develop the system they are looking for.

1) CIS422-424 Capstone Sequence: Systems Analysis, Design and Implementation

The first course in the sequence concentrates on systems analysis and high-level design while the second course concentrates on low-level design, implementation, testing and deployment. A brief summary is presented below of a project that was carried out by the CIS students in their capstone courses for the past two years.

In 2001-2002, the class developed an online ordering system for a radiator manufacturer who builds customized radiators. The students were responsible for developing an online Flash application that would allow a customer to drag-and-drop draw their custom radiator. After the customer had completed the drawing, the specifications were emailed to the manufacturer so that a quote could be created. In addition, the specifications were harvested out of the email and entered into a database so that the particular specifications could be located should additional radiators of that type need to be reproduced. Also, a more customized version of the Flash radiator drawing tool was attached to the in-house database allowing the manufacturer to quickly draw up new radiators. The tool also allowed the manufacturer to view any radiator as a 3-D image from the specifications in the database; in addition a blueprint could be printed for fabricating.

In 2002-2003 the class developed a membership tracking system and point-of-sale system for the local YMCA. This project involved learning how to perform serial communications to work with barcode scanners, electronic door-strikes, CCD cameras, specialized printers (ID card and receipt) and cash drawers. Almost no students had any real hardware experience prior to this course; many left completely understanding how to perform lots of serial I/O coding. In addition, the project involved storing the information into an Access database on a Windows 2000 server and making that data available across a VPN to satellite locations. The students were responsible for locating and setting up the DSL connections, performing wiring modifications at the YMCA campuses and locating and installing the Smooth wall firewall/VPN package on a Debian Linux server.

The students are responsible for carrying out every phase of the project: feasibility studies, requirements engineering, analysis, design, coding, testing, quality control, documentation, training, planning for maintenance, planning for disaster recovery and so on. In short, they see the entire systems lifecycle and are responsible for every aspect of it. The experience for them is simply priceless. Many say they never worked so hard nor learned so much. The majority are grateful for the experience and have commented how much better prepared they were to enter the workforce as compared to their peers from other institutions.

2) CS421-422 Capstone Sequence: Software Engineering

The software engineering option of the computer science degree has a two-course sequence as a capstone. This sequence replaced an earlier single course. We have been running the sequence for two full years and are beginning our third. The first course covers most of the typical software engineering topics with an emphasis on object-oriented analysis and design. During the first course the students are introduced to the project and complete what is the most important deliverable of the project, the design document. During the second course the students must actually create the system promised by the design document. For the project, the students form teams of three to five students. These teams negotiate for different parts of the project. Each team is required to hold regular meetings to facilitate communication between team members.

The project can run on one or more of the systems available to students. One year the project involved a MUD server on a Linux machine written in C++ and a client running on IBM PC's written in Visual Basic. Software tools used by the students have included the group features of Blackboard, e-mail, Visual Studio, and Visio.

Although the students have some programming experience, this is often the first course where they have to work through a whole project in a team. In the second course the students discover the importance of the earlier deliverables, because they use them to answer quite a few questions. Changes in the design documents must be agreed to by the professor and the class as a whole. In the first course the professor tries to evaluate the deliverables carefully and adjust their scope to ensure feasibility. Sometimes this process works well and sometimes non-trivial adjustments must be made in the second course.

3) CS401-402 Capstone Sequence: Computer Networking

The networking capstone courses are developed for students to showcase their creativity and initiative. They are truly a marvelous opportunity for them to develop further their independent study and technical skills. The soft skills acquired are also immense to do primarily with time management, setting up and focusing on goals and targets and harnessing a sense of achievement. The capstones are geared towards creating a synthesis of their previous knowledge; so that they will provide the necessary impetus to elevate and adorn their fresh aspirations in becoming the future academicians but also practitioners.

These experiences are carried out over two terms. They involve both individual and group projects. They account for 30 to 50% of course work. They are mostly dedicated to designing, implementing and testing hardware and software network systems and applications. Unlike the course hands-on labs and programming assignments, these capstone opportunities have to solve a real-world scenario so that the end product is potentially useful. At the end of the projects, the students are expected to submit a written report, and perform a practical demonstration of their precious work before the entire class and the professor.

Some of the projects that were carried out or are currently ongoing are related to: Designing, implementing, configuring and testing a copper, fiber optic and wireless network; Developing ecommerce stores through web programming; Develop secure directory services for organizing information, searching and authentication, using both active directory and openLDAP; Industrial control front end network interface; Developing a secure SMTP server; Developing a Beowulf cluster using both Windows 2000 and Linux; Developing an online cooperative virtual learning system.

These capstone experiences truly generate in the students so much enticement and credence in their abilities to achieve above and beyond any previous experiences they will have had with single program courses. The feedback received from the concerned students and faculty resonates with satisfaction, and that the skills developed will outclass any learning procured with just lectures or reading assignments alone. What is really more important is that these met challenges have all the hallmarks to open for the students windows of reflections into their future career prospects. Their sagacity now seems to tilt from learning outcomes realized in the individual courses to looking forward to real-world openings such as the ones they experienced throughout the capstone projects.

V. SUMMARY

While it is very time-consuming to revamp a curriculum to balance theory and application, deploy technology expecting students to learn from it and manage capstone courses that provide meaningful experiences, we are convinced this is the real mission of undergraduate education. If we want our students to be as prepared as possible to enter the workforce, this is the work we as educators must take on to help them.

The skill sets needed to be successful on the job change yearly and typically academia can be so staunch and slow moving to take them into account. Rather than constantly revising the curriculum, providing courses that have lab components, which can change with the times, is a good choice. To let students know "what's out there" offers capstone projects that partner with industry. If you do these things, then your students are better prepared to hit the ground running when they enter the workforce.

REFERENCES

[1] SM Dascalu, et al., "Computer Science Capstone Course Senior Projects: From Project Idea to Prototype Implementation," Proceeding 35th Annual Conference on Frontiers in Education, Oct. 19-22 2005, pp. S3J- 1-6.

[2] R. Bryant, "The Capstone Course: What Are We Looking for?," Journal of Computing Sciences in Colleges, Vol. 17, Issue 2, December 2001, pp. 46-46.