

CREATING A COURSE IN BUSINESS SYSTEM INTEGRATION: PROCESS, CHALLENGES, AND LESSONS

Vic Matta, Ohio University , matta@ohio.edu,
 Laura Snitcher, Progressive Insurance, Laura_L_Snitcher@progressive.com,

ABSTRACT

Integration of courses across the Management Information System (MIS) curriculum and their relevance to industry has been a significant topic of discussion in IS departments for the past few years. To address this and to respond to declining enrollments, many departments have made major curricular changes. Some of these involve redesign of individual courses, while others require creation or deletion of complete courses. This paper presents a case study of the creation of a new senior level course in Business Systems Integration at a medium sized Midwestern university. Course evaluations and focus groups revealed that the first iteration of this course lacked cohesion and organization. This led to a well informed and researched process of redesigning the course to mend its problems. Driven by MSIS2006 curriculum and classic project-based learning and pedagogical directives, a second iteration was created that resulted in a more streamlined and organized course. This paper outlines the entire process from creation of the course to the final metamorphosis that took place for its adoption as a successful course. It also summarizes the problems faced in development and progression of the course, and presents pedagogical lessons learned.

INTRODUCTION

As instructors who teach Information Systems (IS), we at Ohio University have been increasingly focusing on issues of integration and of relevance of our curriculum to industry. To address these two issues, we felt our current course structure needed to be revamped. Created out of evocations from the MIS advisory board, student evaluations, and focus groups comprising of current students, alumni, and MIS faculty, we made two significant changes to the curriculum. In addition to redesigning our curriculum (shown in Table 1), we also created a senior course in Business Systems Integration (BSI) with the overarching purpose of integrating the IS concepts learned in context of a programming language – C#. This article focuses on this course.

Table 1. Revamped MIS Curriculum

| | MIS Course Sequence |
|----------------|---|
| 201 | Information Analysis and Design |
| 202 | Business Information Systems (SDLC Project, IS Business Fundamentals and Core Issues) |
| 220 | Application Development (VB.NET) |
| 320/ 380 | Systems Analysis, Design and Development & Database Design and Development |
| 325 | Networking and Hardware |
| 400 | Business Systems Integration (C#.NET) |
| 420/ 485 | Capstone Series (Advanced Systems Development with live client, Systems, & Issues) |
| Elect- ives | XML, ERP/Supply Chain, Java, UML, Adv. ASP.NET, Ajax, Project Management |

As with any pilot course, a few adjustments in concepts and course materials took place over three quarters. Despite these adjustments, the course was still sub-optimal. While the content and its delivery were solid, challenges still lay in learning the integrative aspects of the course. Students struggled to apply previously learned concepts in development and systems analysis to complete projects in the context of C#. In other words, they did not see the whole picture.

We began with a course that had undergone several incremental changes resulting in inconsistent teaching materials, to one that effectively and integratively taught systems analysis and design, database development, programming in C#, and yet provided a project management experience.

In the next section, we present existing literature to corroborate the initiatives we undertook in revamping the BSI course, followed by a brief description of what it was initially. Our discussion then details the problems we encountered while piloting the course, and the process we undertook to fix it. The paper then describes what the course eventually achieved, and concludes by discussing some of the lessons learned, and the challenges faced; and some that still remain.

COURSE DEVELOPMENT

Few professions have seen a change as rapid as in the field of IS and IT. Practitioners and academic researchers agree that job requirements and associated knowledge of IS professionals are changing rapidly [4]. The MSIS 2006 Curriculum [1] proposed changes to reflect the recent trends in technology and business that suggested reducing the emphasis on Information Technology (IT) and hardware/software issues, and upgrading Systems Integration to include business processes and systems analysis and design (SA&D). With these guidelines in mind, the course that emerged not only contained an SA&D component, but also integrated technological concepts, consulting practices, communication and presentation requirements [3; 7]. The course used a teaching approach wherein the students worked inter- as well as intra-collaboratively [2; 6], and utilized teams and SDLC methodologies to complete projects [8].

The Initial Course, its Problems and Challenges

Prior to this class, students had learned basic skills in programming, system analysis and design, and database development. As we stated briefly in the introduction, the initial purpose of the Business Systems Integration course was to allow students to combine all of their previously gained knowledge and apply it in the context of a new language (C#.NET) to create a website for a real-life client [3]. In addition to completing a project, the students were required to complete a weekly homework assignments and quizzes.

The concepts taught during the course included lectures, hands-on development, homework, and quizzes on concepts such as code structures (conditional clauses and loops), Access Data Sources, methods, and web services. Because a course of this nature was not common, there were no books that encompassed these concepts fully. Rather we taught these concepts by the means of in class examples that were drawn from different online and textual sources. The course introduced the system development life cycle (SDLC) approach and provided eight weeks of instruction in programming with C# during which students worked on pre-development phases (current state analysis, requirements definition, and design) of the SDLC. The idea was that students would take these examples and apply the concepts to their projects.

While the first design of the course was made with much consideration, students were dissatisfied. Course evaluations hovered at six points below department average. Some quotes from the teaching evaluations helped us understand the issues, for example:

- "There needs to be more in class assignments for every new topic brought up in class."
- "The in-class assignments should directly be related to the homework's as well."
- "I would like a little more structure further ahead of time."

The problems encountered in the first version of the course are summarized below:

1. The course was unorganized and hard to follow due to continuous improvements made to the course over the past three quarters.
2. Since the course used previously learned concepts, it had a steep learning curve and moved quickly. The students were expected to learn a new programming language (C#), complete homework assignments and quizzes, as well as complete a project.
3. The class used in-class time for writing code to teach the more difficult C# concepts. The development of the code took away from the big-picture concepts being covered. As a result, students did not always have a good understanding the code they were writing as well as the concept they were being taught.
4. Although we were teaching by example, the students were unable to translate that information into what they needed in order to complete their projects.
5. Often, while working with the hands-on code being demonstrated in class, some students would develop errors that would be hard to fix. Occasionally, the students were not able to keep up with the hands-on code. This would cause them to become frustrated, develop a negative attitude and give up. This disruption would also significantly slow down the course delivery.

Fixes to the Issues, and the Resulting Course

After a careful review of issues faced with the initial course, we (a faculty member teaching the course and an MIS student) developed a plan to revamp the entire course. The endeavor consumed over 750 hours between the two and resulted in a new course that addressed all the issues, included a course packet, and a mechanism for organized delivery of the course over the quarter. The development followed the course of action outlined below:

1. **Streamlining information:** We reduced the amount of information the students were being exposed to during the course in a two step process. First, we removed the outdated or repetitive topics. Secondly, we extracted concepts that were taught in other courses. For example, we reduced the focus on the SDLC because the students had already thoroughly covered it in a prior course on Systems Analysis and Design. Additionally, we eliminated the Project Management unit since this was the focus of the student's next course in MIS.
2. **Breaking down complex topics through modularization:** We modularized concepts of the course into week long units. This helped to break down complex concepts into more easily consumable parts which were introduced in a lecture, then practiced using in-class hands-on examples, revisited as a homework assignment, and finally tested in a quiz. This procedure is explained further below.
3. **Revamping course objectives:** The course has three high-level objectives:

- *Integrate technology skills to provide effective solutions for business problems:* This objective was accomplished by teaching technological concepts in a logical progression beginning with fundamental concepts and culminating in web services to solve a business problem, as depicted in Figure 1.
- *Demonstrate effective and practical use of Systems Analysis and Design by using a problem-based learning methodology:* Students began the initial phases of the SDLC in parallel technological instruction. Application development began in the final week; students then presented their project to the client and a panel of judges. This project is a culmination of their technological, SA&D, and communication skills.
- *Develop proficient communication and presentation skills:* Students met with clients to translate business needs into technical functionalities, and presented a completed

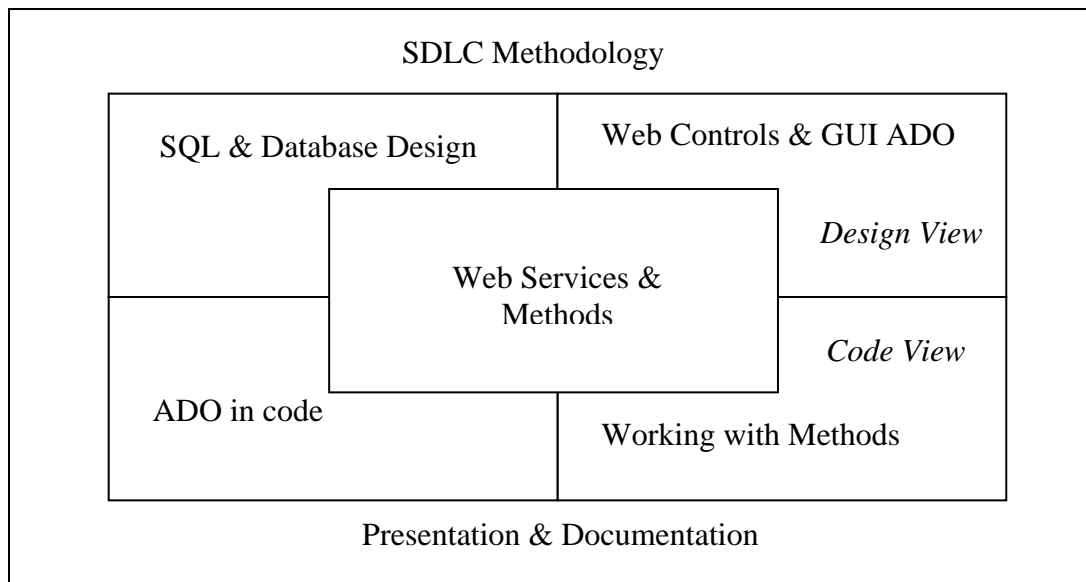


Figure 1. The concept map for BSI

4. **Implementing pedagogical principles:** Any teaching exercise begins with definition of concepts, and then perhaps an explanation. This must be followed by an example showing its use, and finally the student must apply the concept in a specific case. This re-use is critical to completing the learning experience [9]. The following four activities illustrate how this pedagogical style was implemented in the revamped course:
 - *Lecture:* Each module began with a discussion of a concept using PowerPoint. To enhance

and well documented project to the client. learning, students were provided with complete notes on the lecture in the form of a handout optimized for maximizing learning [5]. The notes defined concepts required, for example the definition of DataReaders would be defined by Microsoft in their Microsoft Developer Network (MSDN), and then explained in lay terms. This casual explanation would also show a static screenshot of the instances of its use.

- *Hands-On:* Next a hands-on example would be built in class with the students following each line of code. When the code was

complete, the entire class would run the code, and review its execution. The hands-on examples were created using a project-based learning approach (see the “Refining examples” section below for a better description).

- *Homework*: Students would be assigned homework that would parallel the functionality taught in class while using a different case. This homework required the students to create a professional quality deliverable with specific requirements that would make appropriate reports. Also, a rubric was created for each assignment to identify significant requirements and maintain standardization of scoring.
- *Quiz*: In addition to completing the take home assignment, the students also needed to complete a quiz. This quiz challenged them to again illustrate that they learned the concepts. The quizzes were online and took place in the classroom. Immediately following the quiz, the students were able to see their scores and ask questions. This ensured that they did not move onto the next concept without fully understanding the previous one.

5. Refining examples by adding a project-based learning approach: We improved the example code used for the in-class hands-on exercises so that the students could better see the whole picture and in turn create better websites. A lot of thought was put into how the class could use progressively more sophisticated code without getting more tedious. We created examples that would build off of one another and ultimately create a website that was built around a single business. This worked well because we were able to teach project development using a fictitious groceries business which in turn allowed us to not only include all concepts the class needed to cover, but also allowed the students to see how their code can be used in real world situations. Also, the website is based on a classic customer-product database design that the students were familiar with from previous courses. Re-use of familiar concepts allowed a more solid platform to build on and thereby achieve a higher level of learning. In addition, we developed SDLC documentation corresponding to the business, and introduced it at the beginning of the class to give the students an understanding of what they will be creating during the course. Finally, to reinforce the project-based learning methodology, we framed the homework assignments on another project that had potential interest to students – a membership oriented dating website.

6. Addressing issues regarding falling behind in class: In previous classes, students were having trouble keeping up in class. As we discussed, this was a critical issue because each hands-on web application was based on a previous one, so any incomplete work stalled progress completely. Occasionally, students needed assistance in staying with the pace of in-class hands-on development; either because they preferred to work at a slower pace, had errors, missed class, or were too busy simply documenting notes to be able to keep up with writing code. In order to combat this problem, we created step-by-step directions with screen shots that led students through each in-class exercise one step at a time. This served a number of functions:

- If students fell behind, they could follow the step-by-step to catch up.
- If the student received an error, they could easily go back and see what they did wrong.
- It allowed students who missed class to catch up with the in class activities.
- It gave students a reference as they coded their own projects.

7. Helping students better understand code: Another great feature of the step-by-step instructions was that we included with each step an explanation of what the student was coding and why. We also explained how to read the code. In other words, we would explain an IF loop by stating, “If a person enters a school rank of sophomore then add them to the list, otherwise do nothing.” This helped students to not just be able to write the code, but understand what they wrote.

8. Increasing student confidence by teaching debugging: Although the students understood the concept of debugging, they seemed to have difficulty finding their own errors. By teaching them the simple processes used in debugging by utilizing the Visual Web Developer tools, we taught the students to be more confident in their code. This also helped to increase the morale of the students while also decreasing the amount of time spent helping solve errors.

9. Creating organization: Many of the things described above contribute greatly to the amount of organization in the course. Pedagogical principles we followed were helpful, however the course suffered from lack of delivery vehicles - there was no single text that satisfied the course objectives, and there was no standardized/centralized way to distribute electronic material.

- *Blackboard*: We introduced the use of Blackboard as a centralized one-stop course

management and delivery tool. Lectures, step-by-step for in-class hands-on exercises, and assignments were deployed using Blackboard's course management system. The quizzing facility in particular was helpful by allowing the students to receive instant feedback on their answers. Blackboard was also helpful in creating course announcements and reminders, and posting links to the in-class project, as well as databases used.

- *Printed Course packet:* We created a course packet that followed the structure outlined under pedagogical principles – unit-wise modularization, beginning with concept discussion, a demonstration of problem and solution and ending with the student's application of knowledge in quizzes and assignments.

DISCUSSION

As a result of the course redesign, the course evaluations showed a significant increase in student satisfaction. This is evident in the course evaluations whose summary is outlined in figure 2. In the course evaluation, students were asked to rank the statements on a scale of 1-5 (5 being the highest). As you can see, the average rating in each category increased a significant amount.

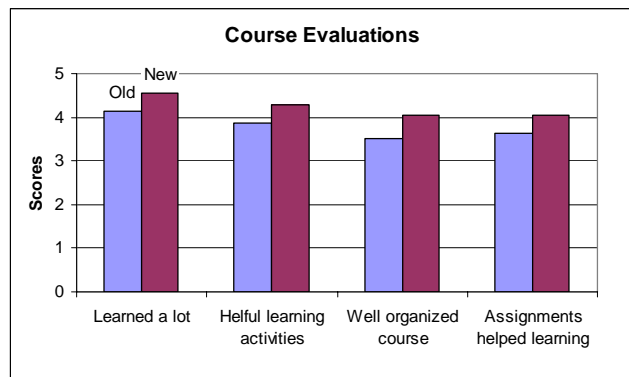


Figure 2. Course Evaluations

The course evaluation also asked students for their comments on the course. Below you will find a list of the comments given by students after the course was revamped:

- "The in class as well as the at home assignments greatly helped my understanding of the course material. Also the group project was great at applying what we learned in class to instances that aren't exactly the same as in class."
- "The instructions and expectation of the class were clear. The book helped reinforce the in class

material. The class was very practical from a development standpoint."

- "I found the in class examples and example web page we created very helpful in my learning."

The class was more organized, and students began to understand all concepts. They were able to apply what was being taught in class to their class projects. Furthermore, there was an increase in student morale because they felt more confident with their coding abilities as they were able to find their own errors.

Challenges that still exist

Although the System's Integration course has improved significantly, some challenges still remain. One challenge centers on the constant struggle with lack of time in a ten week quarter system. Seven to eight weeks devoted to instruction, leaving only two weeks for development, completion and final presentation of project. This often creates stressful situations since the students are not yet comfortable with the concepts they have learned, and have little time to debug their application. Other challenges exist in the form of shortage of unique cases or client projects that would fit well with the course structure. Also, sometimes a university holiday creates scheduling anomalies.

Lessons Learned

After reviewing our approach to solving problems we encountered in the System's Integration course, we found that there are a couple important lessons that we learned that are worth mentioning. First, students tend to respond favorably to the project-based learning approach. In the initial course, it was thought that the students were already learning in a project-based approach because they were expected to complete a project by the end of the quarter. But we found that they had difficulty applying the code learned in class to their own projects. By teaching them how to code in real-life situations and with real life problems rather than hypothetical, they are not only able to learn the coding language, but they are also learning ways in which to use that code.

Another lesson we learned is that lectures and step-by-step coding instruction are insufficient; students need reference materials in order to fully understand concepts. In the initial course, students were given power point slides to introduce a topic as well as code using step-by-step instruction, but they did not leave class feeling comfortable with the concept or the code. Rather they felt lost and unable to catch up.

By adding a course packet that included lecture slides with supporting notes and step-by-step directions of in-class activities, the students were able to go over what they had in class to make sure that they fully understood what was taught. It also gave them a reference when it came time for them to complete their projects.

We also learned that teaching students to develop code was not enough. Students must learn the basics of coding as well as the concepts, but do not need to memorize exact syntax. Rather students must be able to solve problems in their own code and if not able to they must be able to find the information they need to solve that problem. This is why we added lessons on debugging and how to use MSDN. These lectures give the students the tools they need to solve coding problems as they arise.

Finally, we found that by allowing a student to give input to the course creation, we solved many problems that the faculty did not know existed. It is important to note that not just any student will provide good insight, but rather a student who completed and excelled in the IS courses and has had either an internship or real world experience. A student of this kind is able to give a helpful perspective to your course planning. They are able to explain why and how students are getting confused with the current course, clarify what concepts were important and or relevant to their future, as well as point out inconsistencies and redundancies between courses throughout the curriculum.

We continue to refine the course by finding simpler exercises that provide the same outcomes, paring off extraneous steps, and re-writing lectures or step-by-step directions for developing task oriented web applications.

REFERENCES

- [1] Gorgone, J. T., Gray, P., Stohr, E. A., Valacich, J. S., & Wigand, R. T. (2005). MSIS2006 curriculum preview. *Communications of AIS*, 2005(15), 544.
- [2] Hartzel, K. S., Spangler, W., Gal-Or, M., & Jones, T. (2003). A case-based approach to integrating an information technology curriculum. *Information Systems Education Journal*, 1(47).
- [3] Jensen, J., & Wee, L. C. (2000). Creating real life project opportunities for systems analysis and design students. *Information Systems Education Journal*.
- [4] Lee, D. M. S., Trauth, E. M., & Farwell, D. (1995). Critical skills and knowledge requirements of is professionals: A joint academic/industry. *MIS Quarterly*, 19(3), 313.
- [5] Matta, V., Frost, R., & Huang, W. (2005). An exploratory investigation of the effect on learning outcomes of different types of presentation handouts. *Issues in Information Systems*, 6(1), 66-72.
- [6] Nance, W. (2000). Improving information systems students' teamwork and project management capabilities: Experiences from an innovative classroom. *Information Technology and Management*, 1, 293-306.
- [7] Rosenthal, P. (2003). Strategy course and integration course redundancy in the msis2000 model curriculum. *Information Systems Education and Journal*, 1(14), 1-14.
- [8] Steenkamp, A., & Van, D. (2004). An approach to teaching it life cycle processes. *ISECON*, 21.
- [9] Zull, J. (2002). *The art of changing the brain: Enriching the practice of teaching by exploring the biology of learning*: Stylus Publishing.