
EMPOWERING STUDENTS IN THE INFORMATION SYSTEMS CAPSTONE

Kiku Jones, Quinnipiac University, kiku.jones@quinnipiac.edu

ABSTRACT

Over the years, there has been a lot of discussion surrounding the information systems capstone course. Much of this discussion centered on the type of project to be used in the course if one at all. However, there is another element that should be considered: directed vs. non-directed. This paper provides a method of delivering this course that puts the decision-making in the hands of the students. Students are empowered right from the beginning at project selection. This continues throughout the semester as they decide scope requirements and development methodologies. This method of instruction provides a richer experience that students can take with them as they graduate and enter the workforce. The class is structured with project managers, team leaders, and analyst/programmers. The professor fills the roles of CIO, client, and mentor throughout the course. While this is a substantial amount of set-up time for the professor, it ultimately proved to increase the students' confidence and skill sets as demonstrated in their lessons learned comments.

Keywords: Decision-making, IS Capstone, Project Management, Case Study, Information Systems

INTRODUCTION

“Being able to literally create our project and decide what exactly we wanted to accomplish and build was a great experience. Between the research of other companies, and different software that we could use, it really put the experience in our hands rather than just modeling something that already existed or was given to us.” - Student

Various methods for teaching the capstone in information systems have been the topic of many papers [2, 3, 5-9]. Generally, these are discussions of whether to use a real world project or a simulated project; whether to use for-profit or not-for-profit organizations; and even whether or not it makes sense to have a systems development project at all. This paper provides a new dimension to be added to the capstone discussion: empowering students to make the decisions from project selection to scope to development methods. Rather than provide a strict directed method for completing the project, the students play a large role in making these decisions.

Coming from industry, the professor found that one of the biggest issues her employees had was not that they didn't have the requisite skill set, but rather that in an unstructured situation, they lacked the confidence to call on those skills to solve a problem. The students were receiving the proper course work in their programs. However, in each class, they were instructed what was to be done, how it was to be done, and when it was to be done. When these students enter the workforce, these instructions are not always so clear. In those cases, students needed to have the confidence that 1) they have the appropriate foundational skills, 2) they have the ability to research and find any missing pieces, and 3) they are capable of making decisions when it came to selecting the appropriate actions in these unstructured situations. Therefore, the professor felt the best way to develop this confidence and these skills was to provide this experience in the capstone course. The professor sought to answer the question, “do students learn more by being empowered to make decisions regarding their experience in the course?”

In this course, the professor serves as the CIO, the client, as well as the mentor. The first part of the semester begins as any other course with a mix of lecture and activities. However, the real experience begins after the sixth week when the class no longer meets as a group, but rather meets with their respective teams to complete their selected projects each following their own paths.

This paper will provide a discussion of the literature review, project creation, interview and team creation, lectures, milestones, and finally reflections and lessons learned.

LITERATURE REVIEW

There have been many articles discussing what should be included in an information systems course. Gill and Ritzhaupt [2] found that most capstone courses incorporate in some way four themes: 1) a real-world focus, design, development and integration; 2) case methods; 3) using some type of ePortfolio to display their work; and 4) students in the information systems program interacting with professionals in the field. They used these themes in their own development of the capstone course and found that students were generally satisfied; however, there was not a significant increase in the students' critical thinking skills.

Leidig and Lange [6] used community-based non-profit organizations as the focus of their projects in the information systems capstone course. They felt that their students had the appropriate technical skills, but lacked understanding in areas of "organizational processes, team project experience, and the ability to integrate information technology into an organizational setting" (p. 1). They provided lessons learned from reviewing almost 100 projects over 10 years of teaching the capstone course. They reported that working with the non-profits provided many benefits including students receiving in-depth real-world experience and the non-profit organization receiving working software applications that solved a specific need for them.

Reinicke, et al. [8] created their information systems capstone course with an integrated curriculum in mind. They utilized constructivist learning theory which "emphasizes the usefulness of combining and building on previous knowledge that typically happens in a capstone course" (p. 11). In addition, they applied the resource-based view of learning which shifts the role of the instructor to more of a mentor or guide in the learning process rather than a disseminator of knowledge. Ultimately, the authors felt that using an integrated curriculum for the capstone course provided an opportunity for continuous improvement through a two way integration: capstone course providing hands-on learning of concepts from previous courses and the pre-requisite courses providing the appropriate knowledge for students to be successful in the information systems capstone course.

This paper describes a capstone course which includes aspects of each of the above recommended methods of instructing the capstone course. In addition, the course is structured to bring in student-directed learning by empowering the students to make decisions throughout the process. Gibson [1] empowered her students throughout her sophomore-level Sociology of Aging class. She asked her students what they wanted to learn in the course. She provided them with a list of possible objectives and asked them to select four to six. The students were also requested to come up with ways to evaluate whether or not they had achieved the set objectives. Once again, examples were provided from which the students were to select. While she did have a few "non-negotiable" items (i.e., service learning component, keep to the course description, and write in APA style), the majority of the students' paths were decided upon by the students. Comments collected from students after the course were highly positive in favor of the student-directed learning process.

Hains and Smith [4] cross examined eight individual cases of student-directed learning courses to gain an understanding of both student and faculty perspectives. The students developed eight learning outcomes and included a service learning project. The course was titled Cultural and Agrarian Experiential Learning in Agricultural Education. Throughout the course development and course implementation, students were asked to keep personal journals as well as create a course documentary. In the beginning, students were very hesitant. However, as the course continued, the students began to gain more confidence and started to show "more exploration regarding their educational experiences" (p. 368). The below sections will describe how this concept was brought into the information systems capstone course.

PROJECT CREATION

While working with real clients is a great experience for students, this can also have great restraints. There may be strict requirements for specific software, or constraints in user ability to maintain a sophisticated system. These restrictions often leave little room for students to explore untraditional methods of development. Therefore, the decision was made to create scenarios that were based on previously completed projects with real clients to provide a capstone project management experience.

More projects than would actually be needed were created for the semester. This provided the students an ability to choose between projects and have an experience in project selection. For this particular semester, there were three project managers, so nine projects were created to allow each project manager three projects to consider.

The projects were non-profit in nature and the scenarios allowed flexibility in the project scopes. Each potential project had a website and database component. However, beyond this, there was much flexibility in what was to be included in the end product.

INTERVIEW & TEAM CREATION

On the first day of class, students were given information regarding the course content and design. They were provided the role descriptions found in Table 1. The students were asked to set up an interview time, determine which roles they would like to apply for, and told to bring a resume to the interview. They were allowed to “apply” for more than one position. Since there were 27 students enrolled in the course, there would be a total of 3 project managers (PM), 6 team leaders (TL) (2 under each PM), and 18 analyst/programmers (3 under each TL). The professor served as the CIO with each PM reporting directly to her.

Table 1. Role Descriptions

Role	Accountabilities & Authorities
Project Manager	<p>To confer with the Team Leaders in all aspects and decisions with regard to the application, and to act as the management contact for each teams customer.</p> <ul style="list-style-type: none"> • Authority to approve project plans and changes to plans • Provide project status to CIO from each team • Provide project status to customer • Reviews projects with all teams • Provides a summary project plan to the CIO at least monthly • Provides coaching to Team Leaders for social issues within team
Team Leader	<p>Accountable for all aspects of team’s work in terms of quantity, quality and time constraints. Ensures that all required documentation, training programs, help files and project plans are completed.</p> <ul style="list-style-type: none"> • Provides regular project status to Project Manager, schedule to be agreed upon by the PM and TL’s. • Delivery of project on-time and within budget • Provide effective leadership to team • Prepare system and end-user documentation
Analyst/Programmer	<p>Assisting the Team Leader with the analysis and design of the application; also should contribute to the construction of the project plan. Programming the application with respect to tables, forms, reports, etc...</p> <ul style="list-style-type: none"> • Contribute to the project plan • Contribute in the analysis and design phases • Construct appropriate database tables and linkages • Construct web pages and linkages (if applicable) • Write code and test • Design and construct help files • Assist Team Leader in preparation of all documentation

Each student had his/her interview individually with the professor in her office. During these interviews, students were asked behavioral questions that put them in the role for which they were applying. In addition, regardless of

which position they were applying to, students were asked what they wanted to accomplish during the semester (i.e., what skills did they wish to improve or develop).

Initially, many of the students applied for positions that they really did not want. For those students, it seemed to be for one of two reasons. Many students applied for the team leader position because they were concerned about their lack of skills as programmer. They thought that the “analyst/programmer” position indicated that they *must* program. After it was explained that everyone who was not either a team leader or project manager would be an analyst/programmer and that they would then further be able to define their role (as either an analyst OR a programmer), they tended to change their minds and remove their name from the team leader position.

The other reason students applied for a position they did not want was they applied for the project manager or team leader position because they felt they *should* apply for one of these roles. These students felt that the “leadership” roles were something that they were supposed to try to obtain simply because of their titles. However, during the interview process it was clear that these students had a real desire to develop their programming skills – some even indicated a concern that if they got the team leader position, they would not be able to program as much as they would like.

As this class is the last stop, so to speak, before they begin their careers, it was important that the students had the ability to develop the skills they felt they needed. In addition to the interview, students were asked to complete a beginning assessment that tested their current skill level on a variety of items. Included on the assessment were questions dealing with system development processes, entity relationship diagrams, data flow diagrams, and data gathering methods. Also on the assessment were the questions, “What do you hope to get out of this class? What areas do you think you need to improve?” So when determining who would be hired for each position, their responses during the interview process as well as the answers to these questions were mutually considered.

After all interviews were conducted, the professor evaluated each of the students and determined the roles each student would have over the course of the semester. Teams were created in such a way as to make them as equal in skill level as possible. This first began with the selection of PM’s, then TL’s, and finally the remaining students.

Once teams were announced, PM’s were given the ability to reallocate resources within the two teams reporting to them. If they felt that one team was heavy with a particular skill, they were allowed to switch team members. This was a decision they were able to make at any time during the semester. The PM’s were required only to inform the professor (CIO) in their weekly status report when any of these changes occurred.

Once teams were created, they sat together during the lecture portion of the course. Activities were done to build rapport. Discussions in regards to starting fresh and forgetting about classes they had with team members in the past took place. Scenarios from past projects courses were provided illustrating how people you expect to do poorly actually step-up and some that you expect to do well ultimately drop out of the project. The class focused on this being the time for each of them to find what they uniquely bring to the team.

LECTURES

During the first six weeks of the semester, the whole class met together for the lecture portion of the class. This included lectures and activities focused on project management tools and techniques. Students were introduced to items such as scope, work breakdown structures, and timelines. In addition to project management items, topics such as team membership, relationship management, and systems planning were discussed. After the lecture portion of the course completed, the professor shifted into a mentor role for the teams.

MILESTONES

To ensure students were able to complete the projects in the short time frame, milestones were established. This helped the students have some structure in an otherwise unstructured environment. The deliverables for the project were divided into these nine milestones. This sequence and deadline schedule kept the teams moving forward and provided them with enough time to complete the projects. The milestones included:

1. Project Charter & Contract
2. Project scope, WBS, Schedule/Gantt, Dependencies & Resources
3. Data Flow Diagram
4. Entity Relationship Diagram, Data Dictionary
5. Interface design, Navigation design, Input design, Output design
6. Backend Database
7. Frontend Website
8. Project completion with all scope requirements
9. Documentation (including user manuals) & Presentation

As this was to be completed in a semester, there was not much time between each due date. However, many teams were able to complete the first set of milestones quickly. PM's had the decision of whether or not to continue working on the next milestone or taking a break. PM's and TL's also had the ability to turn in a milestone early to be reviewed prior to officially submitting for a grade. This allowed the teams the ability to refine items based on suggestions. However, to have the time to do this, they had to keep working forward without breaks.

Project Selection

Each PM was given three potential projects. They were told that they needed to choose one project for each team (i.e., two of the three projects would be selected). They were given the freedom to make this decision with or without the input from their team members. However, they were told that some of the projects given to the PM were selected specifically because someone on their team had an interest/passion in the area. PM's were instructed on the importance of including team members and building a sense of ownership among them. However, ultimately, it was their decision.

As a part of the lecture portion of the course, project selection techniques were discussed. This included a weighted scoring model. The students were instructed to look at each project in various terms: which was the easiest to complete; which one would help to develop the most skills; which one can we complete with our current skill set; which one would allow us the most flexibility to do some things we've never done before; which one are we most interested in on a personal level; etc. As an added incentive, the students were informed that they would be presenting their projects at the end of the semester to a group of IT professionals from companies in the surrounding area. So it was important that students took that into consideration when they thought about what they might be presenting. They were told that there was not a competition and that each PM had different potential projects from the others.

Client Meeting and Scope Development

Once the projects were selected, the teams or team representatives had to meet with the "client" to ask questions and determine what requirements must be included in the project. The professor served as the client for this portion of the project. Each team had to schedule a time with the professor separately outside of class to complete this meeting. The professor answered questions, but did not lead the students in any particular direction in regards to actual design or development. She simply provided information regarding the problem that the client was having and the main goal of the project.

From there, the students were allowed to add additional features to the scope. The students were instructed to be very careful and thoughtful with the scope as that was what would be used in the final grading of the projects – did they meet all requirements included in the scope.

For this, teams needed to determine what they felt they could accomplish during the short time period of the semester. To this point, the students had never learned to connect a backend database to a website. In addition, most students were unfamiliar with programming languages that allowed them to do some of the things they wanted to add. So they needed to decide whether or not to add the additional features.

After scopes were submitted to the professor, there was discussion of each of the items for clarity. The professor would recommend that some items be removed or altered in some way. However, it was ultimately their decision. When the teams submitted their final scope, it was accepted and from that point forward used in determining the success of the project.

Development

Students were strongly encouraged not to begin coding their databases until they received feedback on their data flow diagrams and entity relationship diagrams. However, once this was done, they were free to begin development. Rather than instruct students with the exact method to use to complete their projects, students were given the freedom to research, select, and use any development methods they wished.

Some teams went straight to researching various programming languages while others looked for a more template based approach. To help students, a couple of crash courses in PHP and database were held during class time. While these courses were not mandatory, they were strongly encouraged. Additional resources were sent to students to use if they chose to do so. As some students had the desire to improve technical and presenting skills, these students were offered the opportunity to teach a crash course on a particular language to the class. For this, they were instructed to make the decision of what would be most useful to the students as a whole and to design their course to fit this need.

During this portion of the course, the professor served as a mentor. The professor met with the PM's on a weekly basis to discuss what was happening with each team. In addition, the professor would randomly select students from each of the teams to meet individually in her office to allow them to discuss how they felt the project was progressing. During these meetings, the students were given the opportunity to "air" any grievances they had with team members, structure, or whatever issues they may be having at the time. This would then lead to a discussion with the TL or PM – providing anonymity for the team member reporting. The professor would mentor the TL or PM to guide them in resolving some of the issues.

While the teams were developing their systems, a few of the teams had to restart for various reasons (e.g., after selecting a template based site, they found out that they were unable to connect a database; they switched from a SQL database to a MYSQL database; etc.). Ultimately, these "restarts" cut the time that they would have had to add additional features to their projects. For some, it caused them to not fully complete the requirements listed in their scopes. These times allowed for the TL's and PM's to really step in and get experience in motivating and encouraging a discouraged group of students.

Because there was no one way for these teams to complete their projects, to help the students, the professor had to get caught up on the tool/language that the students chose to use. This proved to be quite a bit on the professor's part. However, by learning the tools/languages, the professor was able to point students in the right direction when they were stuck. Sometimes this meant sitting side-by-side with the student looking at every line of code.

As the end of the semester was drawing near, teams had to make decisions in regards to what requirements were left and what refinements were needed. The professor reminded the students to continue to review the scope as this was the contract between them and the client.

Presentations

At the end of the semester, IT professionals were invited to attend a presentation of the projects. They were instructed that the course had a focus on project management and decision-making. Students presented their projects and discussed their decision-making process at each step of the project. In addition, they discussed the challenges they encountered, how they overcame those challenges, and things they would have done differently. After the presentations were complete, the IT professionals had the opportunity to ask questions of the teams. In addition, they provided feedback to the professor.

Evaluations

Students were asked to evaluate each other in regards to the project. The professor explained to the students that this was the time to provide their teammates with constructive criticism/feedback allowing them to make adjustments prior to starting their full-time jobs after graduation. The students were told the reports of the evaluations would be given in aggregate and no one would be provided the author of any specific piece of feedback. The team member's portion of the project grade would be based on their evaluations. If they received a 90% average or above, there would be no adjustment to their overall project grade. If they made between an 80% and 90%, there would be a 10% reduction. If they made between a 70% and 80%, there would be a 20% reduction and so on.

However, even with these caveats, students in this course were reluctant to provide appropriate evaluations. Based on the weekly meetings with the PM's and the individual meetings with the students, the professor knew that the 95% to 100% ratings given were inappropriate for many of the students in the course.

REFLECTIONS & LESSONS LEARNED

There were a few areas that the professor will consider changing in the capstone. One would be a formal sign-off process throughout the milestones. At present, there is a formal sign-off at the point of scope, but nowhere else. Even though the point is to allow the students to have the experience of making their own decisions – and even though the professor provided strong recommendations – because the students are still learning, it would be better to force the students to receive that formal sign off before going down the road of using a tool that may not work for their project.

Another area the professor would change is not allowing the crash courses to be optional. Many students did not attend these sessions because they did not feel they would need the information. Later on during the semester it turned out that they could have used the course. So by making those courses mandatory, this would eliminate that issue.

Finally, the evaluation portion of the course needs to be addressed. Students, who clearly should have received a downward adjustment, were not getting that from their teammates. This also robbed students of receiving what could have been valuable feedback.

Students were asked to reflect on their experience by answering lessons learned open ended format questions. The questions and sample responses are below:

1. What types of lessons in regards to teamwork have you learned from your experience in this class?
"I have learned that a team consists of many different personalities. Team organization, structure, and management does not fit neatly into a template." –Project Manager

"I have been in class with all of my team members in previous semesters. In class we did an exercise which resulted in the learning phrase of "removing headbands". "Removing headbands" refers to wiping the slate clean with the perception of an individual. It was important for me to remove any opinions that I had with team members in order to be successful and cohesive team." – Project Manager

"I have learned that having a team with varied skills is often viewed as a positive thing but also learned that sometimes it can damper a team's success and development. This can be a great thing if you monitor it the right way which we struggled to do originally." – Team Leader

"Friendships, however, could get in the way of productivity or confrontation if someone isn't doing their work on time. I learned that sometimes you have to be stern, even with your friends, during meetings that you have to get something finished." – Team Leader

"I learned that in a group setting, patience and carefully listening is paramount when working with a team." – Analyst/Programmer

“This experience has taught me that when the stakes are high and you are in a more professional environment doing real work that teamwork can be a more pleasant experience than I previously thought.”
– Analyst/Programmer

“I have learned that everyone has different skills and it is important to allocate resources according to these skills. We can all help each other in areas we are less comfortable but accountability and resourcefulness is critical to teamwork.” – Analyst/Programmer

“I learned that not everyone gets what they want out of the project because compromises must be made to create a functional system within project duration.” – Analyst/Programmer

2. What types of lessons in regards to leadership have you learned from your experience in this class?

“Essentially this class taught me that leadership must be flexible; leadership must be dynamic and adaptable to the particular team being managed.” – Project Manager

“I’ve learned that an important skill of a leader is to be able to make a final decision. If members of a group are arguing about how something should be done, the leader should evaluate both sides of an argument, formulate a fair opinion, and express that in order to solve the issue.” – Project Manager

“Being chosen as a team leader this semester has taught me more about the importance of team communication and team management skills.” – Team Leader

“I learned that it is important to stress how each member will be held accountable for all of his or her responsibilities even when there is a friendship involved. This accountability needs to be constantly communicated so that the team stays within the time frame of the WBS.” – Team Leader

“To provide upward feedback and ask for some on your own work can be a valuable thing.” – Team Leader

“By experiencing what I did not like as a management style, I can take those tips in order to make myself a better leader.” – Analyst/Programmer

“In regards to leadership I have learned that proper and effective leadership often comes from a leader acting like a team player.” – Analyst/Programmer

“Leaders do not only guide the team through the project and ensure completion, they must keep their mind on the big picture. The step by step processes are focused on by the team members, but the leader must be able to see the project as a whole and how each piece connects.” – Analyst/Programmer

“You do not need to be a team leader or project manager to be a leader. As a programmer or analyst, it is still easy to take point on tasks or meetings without an official title.” – Analyst/Programmer

3. Do you feel that you have improved – whether it be analytical skills, programming, researching, decision making, overall confidence - have improved as a result of your experience in this class?

“As a result of this class I feel that I have improved with decision making and overall confidence.” – Project Manager

“I’ve always been good at managing things and making sure tasks are completed well and on time, but I’ve learned to have more authority in a group setting and really establish a leadership position, even when working with other very outspoken people.” – Project Manager

"I have improved in the ability to make firm decisions." – Team Leader

"My overall confidence as a leader improved knowing our project was completed on time and successfully." – Team Leader

"Going into the project, I didn't have much coding experience, but I quickly learned there are many different ways to build and code a website with a backend database." – Team Leader

"My researching skills have been improved by the fact that I now understand exactly how important it is to do research. I have learned that decision making goes hand in hand with research." – Analyst/Programmer

"Yes! I have learned a completely new language (PHP)! I was able to apply database skills to a real scenario and see how it really works on a live website. I refreshed my HTML skills and became an expert in JavaScript, a language I had barely previously used. Extremely valuable class!" – Analyst/Programmer

"My overall confidence has been improved as well. Before this class I never thought that I would be able to accomplish a task as big as this one." – Analyst/Programmer

"I learned how to weigh each option, think of the overall project, ask other team members' opinions, and then make a decision." – Analyst/Programmer

4. What would you do differently if you had the chance to re-do your project?

"My groups would finish a milestone early in the week and then not meet or complete any other work for the remainder of the week. If we could have been a bit more proactive in the early stages, we may have had more time towards the prototype developmental stages of the project giving us the ability to explore more options." – Project Manager

"After the project selection both teams should have spent more time researching the systems requirements to better selection resources, such as web platforms, domain providers, and backend databases." – Project Manager

"We should have researched more before identifying requirements for the project scope." – Project Manager

"If I had the chance to re-do our project, I would have decided on what software we were going to use earlier on in the semester." – Team Leader

"If I had the chance to redo our project, I would make sure to utilize the planning phases more in depth." – Team Leader

"If I had the chance to redo my project I would definitely focus more energy on research before any prototyping began." – Analyst/Programmer

"If I had the chance to redo the project I definitely would have put more thought into the planning phase and the diagrams so making the website and databases would have been a lot easier." – Analyst/Programmer

CONCLUSIONS

The presented structure to the information systems capstone course empowered students to complete a project from selection to delivery on their own. The professor served in the roles of instructor, client, CIO, and mentor. While this added additional efforts on the part of the professor, it is clear from the students' responses to the lessons learned questions that it was a huge benefit to them. Even though there is some continuous improvement to do to the course,

the professor will continue to provide this opportunity to the students. This method improves the student's skill sets as well as provide them the confidence they need to enter the workforce.

REFERENCES

1. Gibson, L. (2011). Student-Directed Learning: An Exercise in Student Engagement. *College Teaching*, 59, 95-101.
2. Gill, T. G. and Ritzhaupt, A. D. (2013). Systematically Evaluating the Effectiveness of an Information Systems Capstone Course: Implications for Practice. *Journal of Information Technology Education*, 12, 69-94.
3. Gupta, J. and Wachter, R. (1998). A capstone course in the information systems curriculum. *International Journal of Information Management*, 18(6), 427-441.
4. Hains, B. J. and Smith, B. (2012). Student-Centered Course Design: Empowering Students to Become Self-Directed Learners. *Journal of Experiential Education*, 35(2), 357-374.
5. Keller, S. and Lamp, J. (2010). Open-source: solution to the information systems capstone course dilemma. *Open-source solutions in education: theory and practice*, 107-130.
6. Leidig, P. M. and Lange, D. K. (2012). Lessons Learned From A Decade Of Using Community-Based Non-Profit Organizations In Information Systems Capstone Projects. *Proceedings of the Information Systems Educators Conference* ISSN, 1435.
7. Moshkovich, H. M. (2012). Development Project vs. Research Project in an MIS Capstone Course: Student Perceptions. *Review of Business Research*, 12(2).
8. Reinicke, B., Janicki, T. and Gebauer, J. (2013). Implementing an Integrated Curriculum with an Iterative Process to Support a Capstone Course in Information Systems. *Information Systems Education Journal*, 11(6), 10-17.
9. Wilcox, E. and Zigurs, I. (2003). A Method for Enhancing the Success of Service-Learning Projects in Information Systems Curricula. *Information Systems Education Journal*, 1(17), 3-17.