MAKING OOA REAL – ITERATIVE AND INCREMENTAL PROJECTS

Ted J. Strickland Jr., University of Louisville, ted.strickland@louisville.edu

ABSTRACT

Project-based learning is often used to teach analysis and design in one course, followed by physical design and implementation in the second course. Factors considered in redesigning the systems analysis course for this setting are described in this paper. Approaches to teaching the revised course resulted in higher quality system specifications and a redistribution of the student's workload across the two courses.

Keywords: project-based learning, information systems curriculum development, systems analysis and design.

INTRODUCTION

Courses that teach students the techniques of systems analysis and design, system implementation, and project management are fundamental to the information systems curriculum. The IS Model Curricula '02 recommends topic coverage for these areas as model courses IS 2002.7, IS 2002.9, and IS 2002.10 [9].

The IS Model Curricula '02 is flexible regarding course structure and how courses satisfy degree requirements. Information systems programs may craft unique solutions based on school missions and institutional constraints. The Computer Information Systems (CIS) program at the University of Louisville requires that CIS students complete two courses to fulfill these learning objectives: Systems Analysis and Design (CIS 320, first course) and CIS Development Project (CIS 420, second course). An industry-based information systems development project is shared across the two courses, with analysis and design activities conducted during the first course, and system implementation as the focus for the second course. Project management concepts are applied in both courses. Information systems development projects are selected from local organizations.

A two semester course sequence that applies systems analysis and design techniques to industry or simulated settings is common among information systems programs. The benefits of exposing students to project environments have been described, and guidelines for instructors conducting project-based courses are available [3, 5, 8, 12, 13].

Over several projects, CIS instructors and students observed that a disproportionate amount of the development activity was performed during the second course. It became clear that the system specification created in the first course did not provide an accurate representation of system needs. In Fall 2005, the systems analysis and design course was redesigned to produce higher quality system specifications at the end of the course. Initiatives undertaken to create a learning environment that allowed students to make substantial progress in creating complete and thorough systems analysis deliverables are described in this paper.

COURSE DESIGN – STRATEGIC DECISIONS

Four high-level decisions were viewed as crucial to maintaining the structure of the two course project-based learning experience.

Select Appropriate Projects

Aligning the project with the learning objectives of the CIS program resulted in identification of six characteristics of candidate projects.

1. Easily understood problem – Novices may grasp the fundamentals of the business processes without prior knowledge of the industry.
2. Bounded problem scope – the problem is small; i.e., teams of four to five students may conduct analysis and design activities in one semester followed by one semester for system implementation.
3. Flexible business process – the business process may be modified or redesigned using information systems, especially the Internet.
4. Flexible time frame – The project schedule allows two semesters for development activities prior to system delivery.
5. Web-based solution – users will access the information system through a browser, and a web server is part of the system architecture.
6. Database accessed and updated – business data are stored in a relational database.

Emphasize an Object-oriented Systems Development Methodology

The CIS program emphasizes the identification and development of web-based solutions. A strong object-orientation is fundamental to this focus. A systems development methodology that supports object-oriented analysis (OOA), object-oriented design (OOD), and object-oriented programming (OOP) was chosen to build upon programming topics taught in previous CIS courses. The selected methodology was derived from the Unified Process (UP).

Adopt Textbooks that Balance Breadth with Depth

Experience with course projects indicated that while the breadth of academic textbooks was sufficient, they lacked depth in several areas, especially detailed examples of OOA and OOD modeling using the Unified Modeling Language (UML). To provide depth and additional UML examples, trade press books were adopted [1, 6]. They complemented the academic textbook [7], which provided students with a broad introduction to systems analysis and design concepts. The selection of the different book styles was a tradeoff of depth of coverage and ease of reading. Students often found the trade press books more difficult to comprehend, but they found the examples and guidelines were beneficial in constructing meaningful OOA models using UML.

Determine an Appropriate Project Transition Point between the Two Courses

The four phases of the Unified Process provided a transition point that was aligned closely with the academic textbook. The Inception and Elaboration phases were covered in the first course, and the Construction and Transition phases were covered in the second course. An Elaboration Phase Specification defined the transition point between courses, and it contained all systems deliverables created during the first course.

COURSE DESIGN – TACTICAL DECISIONS

From these strategic course design decisions, teaching approaches that offered higher quality systems analysis deliverables were proposed and adopted. In four offerings of the course since Fall 2005, the following tactics have proven effective.

Focus on the Project Beginning with the First Day of Class

Build excitement for the project and the learning opportunities it provides. The course syllabus emphasized the project milestones, the deliverables, and the processes that guide group activities. The course schedule indicated that one or two class periods per iteration (total of six iterations) were reserved for presentations to the client and for technical reviews with the project manager (instructor). Groups were formed during the first week. The client was invited to class during the first week to introduce the problem and to describe her vision of the new system. As details of the project emerged, project examples were included in the class discussions of OOA/OOD concepts and techniques.

Iterate

Let student teams create the deliverables, and then provide them with the type of feedback that moves them closer to acceptable deliverables. Deliverable expectations were stated, but examples from previous projects were not provided. As a team, students decided the content and format of each deliverable.

Two of six iterations were allocated to the Inception Phase. Each group submitted a deliverable package for Iteration #1. The instructor reviewed it, made written comments, and scored it using a standardized rubric. However, a grade was not assigned. Students were asked to use the feedback in preparing the deliverable package for Iteration #2. It was handled in the same manner – reviewed, commented on, and scored using a rubric, but it was not graded. Next, student teams created the Inception Phase Specification, which included the Iteration #1 and the Iteration #2 deliverables. This deliverable was graded based on the two rubrics used previously. Students were aware of the grading scale. They knew which of the deliverables needed improvement, and what to do to earn the desired grade.

Students benefited from two forms of interaction, as shown in Figure 1. As systems analysts, students used periodic requirements review sessions with the client to verify that system requirements had been captured. Comments from the client corrected misunderstandings and refocused expectations. Also, the client frequently commented on presentation “image”; e.g., professional dress, eye contact and engagement with the client, quality of the slides, and heavy use of technical terms. The project manager (instructor) refrained from commenting during the students’ discussions with the client.

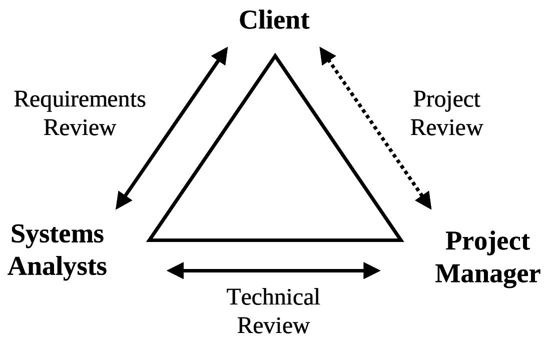

Figure 1. Communication Pathways

During subsequent class periods, students conducted technical review sessions with the project manager (instructor). The project manager's comments focused on addressing two concerns: (a) the quality of the deliverables (based on the rubrics) in modeling the problem domain; and (b) the manner in which students interacted with the client to develop a professional relationship that moved the project forward. Following the technical review sessions, the project manager and client discussed privately the status of the project and expectations for the next round of review sessions (dotted arrow in Figure 1).

Conduct Analysis in Increments

Posture the deliverables as versions, where part of a model is submitted at its due date, and a more detailed version of the model is submitted later. This approach was beneficial to students during the early stages of the project when the vision document, system requirements, and use cases were created.

A draft of the vision document was included in the Iteration #1 deliverable packet. Often, several sections were incomplete or missing. Two weeks later, a completed vision document was included in the Iteration #2 deliverable package. As students interacted with the client and discovered her expectations, they added details to the vision document. Similarly, a list of high level requirements was expanded to form a complete hierarchy of requirements in latter iterations. Initially, use cases were listed as name, purpose, and actor. Later, when the set of use cases was verified with the client and project manager, additional use cases elements (pre-conditions, post-conditions, main flow, alternative flows, etc.) were added.

Prototype

Engage the client in terms that allow her to respond to misunderstandings of expectations. Over several projects, the course instructor observed that clients are most responsive to the following models (in order): (a) high-level prototypes of the "To-be" processes; (b) activity diagrams (flow charts) of the "To-be" processes; (c) use case diagrams and/or use case main flows; (d) formal lists of system requirements; (e) vision document; (f) sequence diagrams and/or class diagrams; and (g) any type of system architecture model and/or design model.

Many clients had mental images of how they expected the new systems to work, especially when the systems were to be deployed on the web. Prototypes provided the client with an opportunity to experience the steps of the "To-be" business process and the data needs at each step. Clients quickly identified misunderstandings and proposed corrections. Also, prototyping was a systems analysis skill that students acquired in previous programming courses. Typically, students earned the highest rubric scores on their prototypes. Also, in creating the prototypes, students gained a deeper understanding of process flow and data needs. This knowledge was useful in creating use cases, sequence diagrams, and class diagrams.

Assess Contributions

Verify that each student is contributing to the group effort. Often students did not put comparable levels of effort into the project. As project manager, the instructor was responsible for identifying students who were not contributing, and for either challenging them to re-engage or eliminating them as obstacles to group success. Peer evaluations were useful in identifying non-performers. They were conducted three times per semester. Decisions based on their results were substantiated with other information. For example, students who did not participate in the requirements reviews with the client or who could not answer questions adequately during the technical reviews usually were not strong contributors to the team.

Additionally, a second procedure for assessing each student's ability to contribute to the project was introduced. After use cases were stable, they were allocated among members of the group. An individual modeling assignment required that each student produce use case descriptions, a use case diagram, sequence diagrams, and a class diagram for the set of use cases assigned to him. These models are the heart of OOA. Success in creating these models was essential to producing a high quality Elaboration Phase Specification. Students who scored well on this assignment were able to combine their individual models in forming the group's models. Students who struggled with this assignment demonstrated knowledge and/or skill gaps that made it difficult for them to contribute to the group effort. In many cases, student performance on the individual assignment confirmed the instructor's observations and substantiated peer evaluation ratings.

Link Exams to the Project

Communicate the need for a common vocabulary and a conceptual understanding when conversing with the project manager (instructor). The project provided an opportunity to develop and to sharpen skills, but it presumed that every student had a grasp of the fundamentals. Students needed to understand that exams and the project were working towards the same end. To make this connection, several exam questions asked students to apply concepts to the project setting. For example, a cost/benefit analysis question asked students to list and to describe project-specific development and operational costs or tangible and intangible benefits.

Promote Professional Behavior

Provide opportunities for students to transition from IS amateurs to IS professionals. Project review sessions with the client generated the richest form of professional feedback for students. The client's comments increased in value when they were reinforced by the project manager (instructor). Additionally, peer evaluations were presented as an opportunity to develop the performance evaluation skills expected of project team members and aspiring managers.

Also, students were required to complete three professional development activities sponsored by the college's career center: (a) attend a professional skills workshop (creating a resume, interviewing basics, making effective presentations); (b) attend a business roundtable event featuring employment opportunities with a company; and (c) participate in a one-on-one advising session with a career counselor. Multiple sessions were offered throughout the semester, and students were expected to register for and to complete all three activities by the end of the course.

The professional development activities were introduced during Fall 2006. Some level of student resistance was anticipated, and career center staff members felt that a few of the business roundtable events may not appeal to CIS students. At the end of the course, students rated their experiences with these activities, as listed in Table 1.

Students' reactions were positive for the professional skills workshops and one-on-one advising sessions. Written comments suggested that students gained more from these sessions than they expected. Reactions to the business roundtable sessions were neutral. Written comments listed: (1) the small number of roundtable events targeting CIS students; (2) the times at which they were offered conflicted with other commitments; and (3) one of the larger roundtable events for CIS students did not provide sufficient exposure to IS career tracks. Overall, these experiences were beneficial to student development, with adjustments planned for subsequent business roundtables.

Table 1. Student Ratings of Professional Development Activities

Professional Development Activity: Student Survey Item	Average Rating	Number of Student Ratings				
		Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The workshop session helped me to identify interpersonal and/or professional skills that I need to develop.	4.16	5	12	2	0	0
The business roundtable session helped me to understand what companies expect of college graduates who wish to work for them.	3.11	0	9	5	3	2
The one-on-one discussion helped me to identify interpersonal and/or professional skills that I need to develop.	4.47	11	6	2	0	0

Rating Scores: Strongly Agree = 5; Agree = 4; Neutral = 3; Disagree = 2; Strongly Disagree = 1

LESSONS LEARNED

As each of the approaches to teaching object-oriented analysis was implemented, surprises were discovered and obstacles were encountered. This section describes the important lessons learned and the issues that still need to addressed.

Surprise: Rubrics Are Your Friends

Before course redesign, evaluating the quality of deliverables was a difficult task for the instructor. Deliverables were based on the students' perceptions of project objectives; i.e., each group envisioned a similar, but different, representation of system needs. Over time, the instructor developed four criteria important to each deliverable: (a) completeness; (b) consistency; (c) communication value; and (d) credibility. Completeness and consistency are comparable to syntactic quality and semantic quality [4]. Rubrics were created to assess performance indicators within each criterion, with a separate rubric for each analysis and design model.

Use of the rubrics had two immediate benefits: (a) evaluation of project deliverables was more thorough, consistent, and timely; and (b) specific feedback was provided to students regarding their deficiencies. The types of improvements needed, and the magnitude of each deficiency, were clear from the rubric. Additionally, the rubrics provided a mechanism for conducting assessments of student learning [2, 10, 11]. The CIS faculty plans to use the rubrics listed in Appendices A through D to conduct embedded assessments of students' modeling skills (use cases, use case diagrams, sequence diagrams, and class diagrams). This assessment will be conducted from student submissions of the individual modeling assignment described above. Combining assessment of student learning with regular class assignments is intended to simplify assessment activities.

Surprise: Prototyping Improves Students' Capture of Requirements

After course redesign, gains in soliciting complete and accurate requirements were attributed largely to improved interaction between the client and students. Prototyping led to most of the improvement. This technique provided an effective means for the client to respond to the students' concepts and to present her ideas.

Students were able to channel their web development skills into building useful representations of the "To-be" business processes. The series of web screens captured the steps of each process, and the data items shown on each screen indicated data items to display or to capture. In many cases, the data items were grouped to form problem domain classes. For some students, the prototypes provided high fidelity realizations of use cases, with sufficient detail to generate the corresponding sequence diagrams and class diagrams. The ability to work from a prototype to a UML model of the system demonstrated a deeper understanding of systems modeling.

Obstacle: Systems Thinking Is Difficult for Students

For many junior-level students, the systems analysis and design course was the first one where critical thinking was required. They struggled with connecting concepts across chapters of the texts and reasoning through cause and effect relationships. They could not visualize how different models of the system represented the same business problem; i.e., how the modeling pieces fit together. For instance, students did not connect system architecture options with the cost/benefit analysis. It seemed that from the student's perspective, the two topics were presented at different class sessions and in different chapters, so how could they be related? Similarly, students translated use cases to sequence diagrams, and independently they translated the same use cases to a class diagram. However, objects on the sequence diagram may not have corresponding classes on the class diagram. These examples typify the difficulty students had learning to think at the systems level.

Obstacle: Students Are Hesitant to Write Descriptions of their Models

Students were slow to recognize that high quality system specifications contain well-crafted models and well-written descriptions of those models. It was not uncommon for students to include sequence diagrams, class diagrams, and deployment diagrams in their deliverable packages, all without one word of explanation. It appeared that students perceived that the objective was to demonstrate the ability to create the model, and they failed to realize the importance of making the models understandable to the reader. When students provided descriptions, they often were not relevant. For example, instead of describing why the database model was required to be designed for third normal form, students listed the rules of normalization. The writing issue appears to be both informational (knowing what is required and why) and motivational.

CONCLUSION

The systems analysis and design course redesign initiatives were successful in improving the quality of the system specifications created in the first course of the two semester course sequence. The instructor of the second course, CIS Development Project, reported only minor changes in system scope and few revisions of system requirements during the implementation phase of the project. Systems development effort across the two courses became more evenly distributed. Also, gains in students' knowledge of UML and their modeling skills were reported.

The teaching approaches adopted for the systems analysis and design course created an environment where students were able to create high quality deliverables. The three teaching approaches that had the largest impacts were: (a) prototyping; (b) iterating to create deliverables; and (c) conducting analysis in increments.

REFERENCES

1. Arlow, J., Neustadt, I. (2005). *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*, *Second Edition*, Upper Saddle River, NJ: Addison-Wesley.
2. Blandford, D.K., Hwang, D.J. (2003). Five Easy but Effective Assessment Methods, *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE* '03, 41-44.
3. Brewer, J., Lorenz, L. (2003). Using UML and Agile Development Methodologies to Teach Object-Oriented Analysis & Design Tools and Techniques, *Proceedings of the 4th Conference on Information Technology Curriculum, CITC4 '03*, 54-57.
4. Bolloju, N., Leung, F.S.K. (2006). Assisting Novice Analysts in Developing Quality Conceptual Models with UML, *Communications of the ACM*, 49(7), 108-112.
5. Clark, N. (2005). Evaluating Student Teams Developing Unique Industry Projects, *Proceedings of the 7th Australasian Conference on Computing Education, ACE '05*, 21-30.
6. Conallen, J. (2002). *Building Web Applications With UML, Second Edition*, Boston, MA: Addison-Wesley.
7. Dennis, A., Wixom, B.H., Tegarden, D. (2005). *Systems Analysis and Design with UML Version 2.0: An Object - Oriented Approach, Second Edition*, Hoboken, NJ: John Wiley & Sons.
8. Fernandez, E., Williamson, D.M. (2003). Using Project-Based Learning to Teach Object Oriented Application Development, *Proceedings of the 4th Conference on Information Technology Curriculum, CITC4 '03*, 37-40.
9. Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., Longenecker, H.E., Jr. (2002). IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery, Association for Information Systems, and Association of Information Technology Professionals.
10. Maxim, B.R. (2004). Closing the Loop: Assessment and Accreditation, Journal of Computing Sciences in Colleges, 20(1), 7-18.
11. Sanders, K.E., McCartney, R. (2003). Program Assessment Tools in Computer Science: A Report from the Trenches, *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '03*, 31-35.
12. Strickland, T.J., Jr. (2003). Real Projects, Real Lessons – What Students Learn and What the Instructor Has Learned, *Issues in Information Systems*, IV(2), 693-699.
13. Tan, J., Phillips, J. (2003). Challenges of Real-World Projects in Team-Based Courses, *Journal of Computing Sciences in Colleges*, 19(2), 265-277.

Appendix A. Use Case Rubric

Evaluation Criterion	Not Present (0)	Not Good Enough (1)	Good Enough (2)	Way Good (3)
Complete #1	Process steps are not included for the use cases.	One or more of the use cases capture **far less than** 80 percent of its process steps.	Each use case captures **approximately** 80 percent of its process steps.	Each use case captures **in excess of** 80 percent of its process steps.
Complete #2	The traceability matrix is not included.	The traceability matrix **fails to** relate use cases to system requirements adequately.	The traceability matrix relates use cases to system requirements **adequately**.	The traceability matrix relates use cases to system requirements in **an outstanding manner**.
Consistent #1	Use case names are not included.	**Far less than** 80 percent of the use cases are named appropriately.	**Approximately** 80 percent of the use cases are named appropriately.	**In excess of** 80 percent of the use cases are named appropriately.
Consistent #2	Use cases are not included.	**Far less than** 80 percent of the use cases conform with the actor/system response format (Arlow and Neustadt).	**Approximately** 80 percent of the use cases conform with the actor/system response format (Arlow and Neustadt).	**In excess of** 80 percent of the use cases conform with the actor/system response format (Arlow and Neustadt).
Consistent #3	Use cases are not included.	**Far less than** 80 percent of the use cases are written in the SVDPI format.	**Approximately** 80 percent of the use cases are written in the SVDPI format.	**In excess of** 80 percent of the use cases are written in the SVDPI format.
Consistent #4	Use cases are not included.	As a set, the uses cases **are not** constructed at a comparable level of detail.	As a set, the uses cases are **generally** constructed at a comparable level of detail.	As a set, the uses cases are **clearly** constructed at a comparable level of detail.
Communicates	Use cases are not included.	Writing errors **are sufficient to distract** attention from the key elements of the use case(s).	A few minor writing errors **do not distract** from the key elements of the use case(s).	**No writing errors** are present.
Credible #1	Use cases are not included.	Overall, the use cases **fail to** demonstrate **a fundamental** understanding of systems modeling using use cases.	Overall, the use cases demonstrate **a fundamental** understanding of systems modeling using use cases.	Overall, the use cases demonstrate **an exceptional** understanding of systems modeling using use cases.
Credible #2	Use cases are not included.	Overall, the use cases **fail to** demonstrate **a fundamental** understanding of the required system functionality.	Overall, the use cases demonstrate **a fundamental** understanding of the required system functionality.	Overall, the use cases demonstrate **an exceptional** understanding of system functionality.

Appendix B. Use Case Diagram Rubric

Evaluation Criterion	Not Present (0)	Not Good Enough (1)	Good Enough (2)	Way Good (3)
Complete	The use case diagram is not included.	The use case diagram **fails to provide an adequate** representation of the system boundary, the actors, the use cases, or the relationships among actors and use cases.	The use case diagram provides an **adequate** representation of the system boundary, the actors, the use cases, and the relationships among actors and use cases.	The use case diagram provides an **outstanding** representation of the system boundary, the actors, the use cases, and the relationships among actors and use cases.
Consistent 1	The use case diagram is not included.	The use case diagram is **not** consistent with the Arlow and Neustadt format.	The use case diagram is **generally** consistent with the Arlow and Neustadt format.	The use case diagram is **entirely** consistent with the Arlow and Neustadt format.
Consistent 2	The use case diagram is not included.	The use case diagram is **not** consistent with the set of use cases.	The use case diagram is **generally** consistent with the set of use cases.	The use case diagram is **entirely** consistent with the set of use cases.
Communicates 1	The narrative is not included.	The narrative **fails to provide an adequate** explanation of the use case diagram in terms the client would understand.	The narrative provides an **adequate** explanation of the use case diagram in terms the client would understand.	The narrative provides an **outstanding** explanation of the use case diagram in terms the client would understand.
Communicates 2	The narrative is not included.	Writing errors **are sufficient to distract** attention from the key points.	A few minor writing errors **do not distract** from the key points.	**No writing errors** are present.
Credible 1	The use case diagram is not included.	Overall, the use case diagram **fails to demonstrate a fundamental** understanding of systems modeling.	Overall, the use case diagram demonstrates **a fundamental** understanding of systems modeling.	Overall, the use case diagram demonstrates **an exceptional** understanding of systems modeling.
Credible 2	The use case diagram is not included.	Overall, the use case diagram **fails to demonstrate a fundamental** understanding of system scope for this stage of systems development.	Overall, the use case diagram demonstrates a **fundamental** understanding of system scope for this stage of systems development.	Overall, the use case diagram demonstrates **an exceptional** understanding of system scope for this stage of systems development.

Appendix C. Sequence Diagram Rubric

Evaluation Criterion	Not Present (0)	Not Good Enough (1)	Good Enough (2)	Way Good (3)
Complete 1	The sequence diagrams are not included.	**Far less than 80 percent** of the use cases have sequence diagrams.	**Approximately 80 percent** of the use cases have sequence diagrams.	**In excess of 80 percent** of the use cases have sequence diagrams.
Complete 2	The sequence diagrams are not included.	**Far less than 80 percent** of one or more of a use case's process steps (main flow) are represented on its sequence diagrams.	**Approximately 80 percent** of each use case's process steps (main flow) are represented on its sequence diagrams.	**Approximately 80 percent** of each use case's process steps (main flow) are represented on its sequence diagrams.
Complete 3	The sequence diagrams are not included.	The sequence diagrams **fail to provide adequate** representations of the actors, the objects, and the messages exchanged among objects.	The sequence diagrams provide **adequate** representations of the actors, the objects, and the messages exchanged among objects.	The sequence diagrams provide **outstanding** representations of the actors, the objects, and the messages exchanged among objects.
Consistent	The sequence diagrams are not included.	**Far less than 80 percent** of the sequence diagrams conform with the sequence diagram guidelines listed in Chapters 12 and 13 of Arlow and Neustadt.	**Approximately 80 percent** of the sequence diagrams conform with the sequence diagram guidelines listed in Chapters 12 and 13 of Arlow and Neustadt.	**In excess of 80 percent** of the sequence diagrams conform with the sequence diagram guidelines listed in Chapters 12 and 13 of Arlow and Neustadt.
Communicates 1	The narrative is not included.	The narrative fails to explain **adequately** how the sequence diagram was constructed using either: (1) CRC cards; (2) verb-noun analysis; or (3) UX modeling of the prototypes.	The narrative explains **adequately** how the sequence diagram was constructed using either: (1) CRC cards; (2) verb-noun analysis; or (3) UX modeling of the prototypes.	The narrative provides **an outstanding explanation** of how the sequence diagram was constructed using either: (1) CRC cards; (2) verb-noun analysis; or (3) UX modeling of the prototypes.
Communicates 2	The narrative is not included.	Writing errors **are sufficient to distract** attention from the key points.	A few minor writing errors **do not distract** from the key points.	**No writing errors** are present.
Credible 1	The sequence diagrams are not included.	Overall, the sequence diagrams **fail to demonstrate a fundamental** understanding of use case realization.	Overall, the sequence diagrams demonstrate **a fundamental** understanding of use case realization.	Overall, the sequence diagrams demonstrate **an exceptional** understanding of use case realization.
Credible 2	The sequence diagrams are not included.	Overall, the sequence diagrams **fail to demonstrate a fundamental** understanding of the dynamic behavior of the problem solution.	Overall, the sequence diagrams demonstrate a **fundamental** understanding of the dynamic behavior of the problem solution.	Overall, the sequence diagrams demonstrate an **exceptional** understanding of the dynamic behavior of the problem solution.

Appendix D. Class Diagram Rubric

Evaluation Criterion	Not Present (0)	Not Good Enough (1)	Good Enough (2)	Way Good (3)
Complete	The class diagram is not included.	**Far less than 80** percent of the classes contain class names, data attributes, operations, and relationships.	**Approximately 80** percent of the classes contain class names, data attributes, operations, and relationships.	**In excess of 80** percent of the classes contain class names, data attributes, operations, and relationships.
Consistent 1	The class diagram is not included.	The class diagram is **not** consistent with the class diagram guidelines listed in Chapters 7 and 8 of Arlow and Neustadt.	The class diagram is **generally** consistent with the class diagram guidelines listed in Chapters 7 and 8 of Arlow and Neustadt.	The class diagram is **entirely** consistent with the class diagram guidelines listed in Chapters 7 and 8 of Arlow and Neustadt.
Consistent 2	The class diagram is not included.	The class diagram is consistent with the **far less than 80 percent** of the sequence diagrams.	The class diagram is consistent with **approximately 80 percent** of the sequence diagrams.	The class diagram is consistent with the **in excess of 80 percent** of the sequence diagrams.
Communicates 1	The narrative is not included.	The narrative **fails to provide an adequate** explanation of the class diagram in terms the client would understand.	The narrative provides an **adequate** explanation of the class diagram in terms the client would understand.	The narrative provides an **outstanding** explanation of the class diagram in terms the client would understand.
Communicates 2	The narrative is not included.	Writing errors **are sufficient to distract** attention from the explanation of the class diagram.	A few minor writing errors **do not distract** attention from the explanation of the class diagram.	**No writing errors** are present.
Credible 1	The class diagram is not included.	Overall, the class diagram **fails to demonstrate a fundamental** understanding of structural modeling.	Overall, the class diagram demonstrates **a fundamental** understanding of structural modeling.	Overall, the class diagram demonstrates **an exceptional** understanding of structural modeling.
Credible 2	The class diagram is not included.	Overall, the class diagram **fails to demonstrate a fundamental** understanding of the static structure of the problem solution.	Overall, the class diagram demonstrates **a fundamental** understanding of the static structure of the problem solution.	Overall, the class diagram demonstrates **an exceptional** understanding of the static structure of the problem solution.