

CQI Report: Software, Spring 2024

1 Subcommittee Information

The following courses are part of this subcommittee. A course may have several sections and instructors.

Subcommittee CS Courses	CRN number	Instructor name
CS 3195 Junior Professional Orientation	24283	Daniel Mejia
CS 3331 Advanced Object-Oriented Programming	21123	Oscar Mondragon
CS 3331 Advanced Object-Oriented Programming	23920	Bhanukiran Gurijala
CS 3331 Advanced Object-Oriented Programming	25725	Bhanukiran Gurijala
CS 4310 Software Engineering I	21678	Oscar Mondragon
CS 4311 Software Engineering II	23072	Salamah Salamah

Report date: 9/30/2024

Faculty meeting presentation date: 10/04/2024

2 Summary

Evidence collection for the assessment took place during the Spring of 2023, with individual course reports finalized in the Summer and Fall 2024. The committee convened on September 23rd, 2024, to review the reports and collected data and discuss recommendations for each course.

- CS 3195: All 78 students enrolled in the course on census day passed the course (D is a passing grade for CS 3195). One outcome was not measured (2.e), all other outcomes were met.
- CS 3331: There were three sections with enrolments of 60, 59 and 33 students. All Course Learning outcomes were met in all three sections except for Outcome 1.b () that was not assessed in all three sections.
- CS 4310: There was one section starting with 94 students. All Level 1, 2, and 3 Learning outcomes were met as good or excellent, but Level 1.d as Marginally met. Instructor will consider project work in addition to exam question.
- CS 4311: All 73 students enrolled in the course by census day passed the course (D is a passing grade for 4311). All 22 outcomes were met. One level 2 outcome was met marginally. We suggest breaking some of the outcomes to provide more granularity in assessment results.

The following sections detail the committee's recommendations and follow-up actions for each course. And the links to the full individual course reports are provided at the end of this document for further review.

3 CS 3195 Junior Professional Orientation

3.1 Previous recommendations - Reflection and Actions Taken

- Expand on the use of technology for communication and collaboration
 - **Action taken:** Utilize LinkedIn, Github, Google Virtual Interview
- Continued having student support in the form of an Instructional Assistant (IA)
 - **Action taken:** IA is utilized to assist with attendance, data collection, and data entry

3.2 Spring 2024 - Recommendations for Course Instructors.

- Add time to the course to conduct additional interviews
 - Both Technical and Behavioral
 - Utilize CIC to establish peer technical interviews
 - Utilize online services for behavioral interviews
- Formally gather course artifacts to measure 2.e (Create a professional repository (e.g., Github) to showcase software development experiences)

3.3 Spring 2024 - Recommendations Requiring Departmental Approval

- Move this course to be earlier in the curriculum (e.g., take the course concurrently with CS 2302)
 - Remove the restriction of Junior Level

4 CS 3331 Advanced Object-Oriented Programming

4.1 Previous recommendations - Reflection and Actions Taken

1. Continue monitoring the failure rate (17%);
Action taken: The failure rate for two sections starting with 60 students is less than 10% in Spring 2024; however, the failure rate for one section with 33 students was 24%. Overall, there is a failure rate of **11.5% average** for the three sections.
2. Consider using a project-based learning approach in this course (i.e., a semester long project that builds upon this course)
Action taken: There was a class project for the semester in the three sections. One section used a semester long project, and the other two sections had an 8-week project.
3. Use a software development approach such as waterfall or agile for the course project.
Action taken: The development approach was iterative and incremental. New requirements were given for each of the labs. Students delivered working deliverables with incremental functionality. Students could use previous work to make modifications.

4.2 Spring 2024 - Recommendations for Course Instructors.

1. Course outcome 1.b (waterfall vs. agile development) was not assessed in all three sections. Make sure to measure and assess this outcome. Note: The outcome was covered with lecture and use of class project incremental deliveries.
2. Analyze the causes behind the 24% failure rate in one section and consider implementing targeted interventions to address and reduce the issue proactively.
3. Continue making use of a project-based class with an incremental approach – new requirements for each increment (lab).
4. Continue introducing teamwork in small teams (3) and using a self-assessment form with % for each work category, e.g., pseudocode, code, and models. Each member assesses the other members in the same form. This activity also promotes ethical values such as honesty. In addition, Team members have a mechanism to indicate if a member is not contributing.
5. Continue effective use of TA/IA support to assess each member's contribution to the project deliverables (labs). TA should consider assessing modeling, problem-solving, and coding skills in next semesters. TA should continue checking the team's self-assessment forms and identify students having problems early in the semester.

4.3 Spring 2024 - Recommendations Requiring Departmental Approval

None

5 CS 4310 Software Engineering I

5.1 Previous recommendations - Reflection and Actions Taken

1. Monitor the 2 outcomes that were not met (L1a Sw. Eng principles) and L2e application of the code of ethics).

Action taken: The instructor is now asking students to identify how Sw. Engineering principles are applied for each of the analysis models covered in class. This has been effective (outcome being met) in the previous course deliveries since Spring 2022.

Action taken: A video on ethics, a summary of the CS IEEE and ACM code of ethics, and a class reflection activity have been added to the course (outcome being met). Students read the code of ethics of professional associations and reflect on the expected ethical behavior in their team and the class project. These changes have been effective in the previous course deliveries since Spring 2022.

2. The size of the class, the size of teams, and the number of TAs assigned for this project-based and team-based capstone class should be revised to ensure the quality of its delivery. Consider having a cap of 50 students per section and a pool of instructors.

Action taken: None. CS enrollment continues to grow as well as the CS-faculty/student ratio. Class is being delivered as a single section with a student enrollment between 85 and 110 in the previous course deliveries since Fall 2022.

Monitor the introduction of the Team Software Process (TSP) framework to support scaling team size from 5 to 9 students while increasing individual performance, commitment, sense of belongingness, and work quality. The following TSP principles should be implemented: Role Managers, Weekly Sprint Schedules, Management Status Reports, and Self-directed Teams.

Action taken: The TSP principles have been implemented every semester since Spring 2022. Student feedback on acquiring teamwork skills is good. Customers provided feedback sharing that more students participated during customer presentations, students were more confident, students delivered more functionality, and the quality of the software products increased.

5.2 Spring 2024 - Recommendations for Course Instructors.

1. Outcome 1.d (*Define security design principles and the rule of least astonishment*) was marginally met with 74%. Monitor this outcome by (a) adding an assessment for the SRS Section on *Quality attributes - Security requirements* where students write security requirements for the class project and (b) creating a quiz on security design principles.
2. To ensure the quality of this team-based capstone course, the class size, team size, number of software reports, and allocation of TA should be carefully reviewed. Given that the class size has exceeded 100 students in previous semesters, it is recommended to discuss with the Chair the need for adequate TA support and possibly a pool of instructors.

5.3 Spring 2024 - Recommendations Requiring Departmental Approval

None

6 CS 4311 Software Engineering II

6.1 Spring 2024 - Previous recommendations - Reflection and Actions Taken

1. Monitor the outcomes that were not met (2b. detailed design) or met marginally (2a. architectural design and 2j. analysis of non-functional properties).
 - a. Performance in 2b outcome has improved to **good**.
 - b. Performance in 2a outcome has improved to **good**.
 - c. Outcome 2j from the previous report is now 2h and students' performance in the outcome has improved to **excellent**.

Action taken (outcome 2b): the course introduces the topic of pre and post conditions much earlier in the semester. Students are given a homework assignment and are tested on the topic during exam 2 and the final exam.

Action taken (outcome 2a): The course now includes a team presentation on one of the architecture styles covered in the course (layered, pipe and filter, cloud, etc.).

Action taken (outcome 2j): The course now has a dedicated cybersecurity workshop session highlighting design issues such as buffer overflow and an exam question assessing students' ability to 1) detect some issues in design, and 2) provide mitigation strategies.

2. The course has two required textbooks: one for theory and the other for practice (CRC approach). Consider updating the old CRC textbook or creating supplementary materials, e.g., a CRC handbook or workbook.

Action taken: The course instructors have provided a list of five textbooks to supplement learning.

3. Let the students use the UML notation to document their CRC-based design work; nobody knows or uses the old CRC notation today. And provide more examples

Action taken: The course makes use of multiple UML notations including class and sequence diagrams. Collaboration graphs remain the main modeling diagram for the design, as it provides seamless transitions between the different design activities for the class (CRC to Contract to Subsystems)

4. There is a process mismatch between what students are taught for their project design work (waterfall lifecycle model) and what they do for the actual implementation (incremental and iterative approach). Consider streamlining the two processes, e.g., incorporating the CRC technique into a modern development approach such as Agile development.

Action taken: We worked with the client to delay the start of demos to allow student teams to arrive at a high-level architecture (skeleton design) prior to the implementation. More work is needed to push the start of the demos even further, and to adjust the course schedule to establish a more solid design (possibly with a complete detailed design for one or two paths through the system) prior to the first demo.

5. Work with the CS 3331 and 4310 instructors to strengthen the (individual) practice of UML class diagrams.

Action take: The work within the CQI committee continues to tackle this.

6.2 Spring 2024 - Recommendations for Course Instructors.

1. As a capstone course within the BSCS program, CS4311 requires a significant amount of work on the students' and instructional team's parts. Periodic meetings with the teams are essential to ensuring that individual students contribute to the team's deliverables. This was accomplished through holding weekly meetings with student teams. Those meetings alternated between client demo in one week, followed by a meeting with the TAs the other week.
2. The capstone courses continue to apply a waterfall approach to requirements and design, while 4311 continues to deliver software code in two-week cycles. A possibility to mitigate this conflict is to hold off all code deliverables until the second third of the semester (6-7 weeks from semester's start) while working on delivering a solid system's design at that time.

3. Multiple course outcomes, in particular level 3 outcomes, are evaluated through team deliverables. To maintain integrity of these evaluations, it is important to continue to:
4. Hold weekly meetings with all team members to ensure contributions of all students, and/or
5. Rely on peer evaluations after each major team deliverable

6.3 Spring 2024 - Recommendations Requiring Departmental Approval

- 1) Break outcome 2c “Distinguish between the different levels of cohesion and coupling” to two outcomes for a more fine-grained assessment:
 - a. 2c distinguish between the different levels of cohesion
 - b. 2d distinguish between the different levels of coupling
- 2) Change outcome 3e to support iterative development process
From: “Construct **software** from a detailed design.”
To: “Write **code** based on a detailed design.”

7 Individual Course Outcomes Reports

- [CS 3195](#)
- [CS 3331 \(CRN: 21123\)](#)
- [CS 3331 \(CRN: 23920\)](#)
- [CS 3331 \(CRN: 25725\)](#)
- [CS 4310 \(CRN: 21678\)](#)
- [CS 4311 \(CRN: 23072\)](#)