Capstone I AY16 Program Review: Program Outcomes and Capstone Assignment Alignment

COMPUTER SCIENCE

Program Learning Outcomes (PLOs)

All program graduates will demonstrate achievement in the following areas:

- Content Knowledge: demonstrate factual and conceptual grasp of the field of computing.
- Integrated Learning and Communication: demonstrate the ability to incorporate learned skills to design, develop, and evaluate software systems of varying complexity to meet desired user requirements.
- Problem Solving: demonstrate proficiency in using one or more industry-standard programming languages and scripting languages to solve problems;
- Inquiry, Critical Thinking, and Analysis: demonstrate ability to apply conceptual knowledge for analysis and problem solving.
- Teamwork and Civic Engagement: demonstrate teamwork ability to work collaboratively with end users and other developers.

Capstone Assignment

CS 401 Capstone Syllabus Excerpt: The Prompt

About this Course

The course allows students to integrate their academic work in the design and development of a significant product that showcases the student's skills. The task of designing and developing software typically requires a considerable amount of time, so students will normally take three terms to complete this course.

Outcomes

Students enrolled in this class will apply their acquired software development, interface design, programming, interpersonal and writing skills to create a finished set of design documents and a first-release version of a software product. When they have completed this course students will have:

1. developed and refined a preliminary proposal;
2. developed and refined complete functional specifications;
3. outlined all screen layouts (as applicable);
4. diagrammed entity relationships, data flow, and other design details (as applicable); and
5. created and tested the initial version of the designed product.

Closing the Loop Statement

Except for Teamwork and Civic Engagement, program learning outcomes are reflected in the assessed outcomes for the capstone project. Capstones are longer-term projects that do not always lend themselves to collaborative work. However, earlier courses in the CS curriculum, beginning with CS 121, have components that address collaboration and teamwork.

Program outcomes and the capstone: The computer science capstone is a design and development project that requires students to create a new software product to address some need, real or perceived. Students must demonstrate thorough understanding of every phase of software design, and select appropriate development tools to create and test a working prototype, or if possible a finished product ready for distribution. Completing this task requires use of content knowledge from across the CS curriculum, the ability to apply the conceptual knowledge to solving the problems embedded in the umbrella task selected for the project (including sometimes extensive revision) and to use the standard tools of the trade (programming languages, database systems, and interactive development environments) to bring their project to fruition.

Teamwork and Civic Engagement and the CS degree program: Software developers often work on projects in teams, and as developers gain seniority and experience, frequently must work closely with end users. However, although teamwork skills are important, it is also critical that students in the curriculum gain thorough knowledge of the concepts and skills required for the job. Satisfying both of these needs requires the curriculum to provide opportunities for collaborative work that don't also provide ways for students to "ride someone else's coat-tails" to degree completion. The use of collaboration in our courses has been in development for the past five years, and presently is represented by the following activities:

CS 121—Introduction to Software Development. Team research and presentation assignment. Teams of two to four students research and prepare short presentations on topics applicable to course content.

CS 161, 162—Foundations of Computer Science. Team Programming "Contests." In teams of two, students solve (or try to solve) programming problems scaled for completion in one class period or less. This is an activity that students seem to enjoy, and we are beginning to introduce the concept in other classes that stress programming skills.

CS 315—Software Design. Group review and feedback of student design document drafts. This is an activity very like that faced in the software development workplace, in which ideas are shared and refined. This class spends about ten percent of its face time engaged in these reviews.

CS 380—Software Engineering. Team software engineering project. This course underwent a renaissance in 2017 and began to address the material by having students work together to develop a problem solution from specifications to module design.

CS 409—Practicum. Internship with "real" people. Internship opportunities are not numerous enough to make this course a required part of the curriculum, but students who are able to take advantage of the opportunity must work with real clients who present them with very real problems to solve.

Collectively the listed opportunities form the mainstay of teamwork and civic engagement in the present curriculum. As yet, formal assessment of these activities is uneven, a shortcoming the program seeks to address.

Action Plan: Create a rubric for Civic Engagement.