August 23, 2020

CEN 5011: Advanced Software Engineering

Fall 2020
Monday/Wednesday, 5:00 PM – 6:15 PM, Zoom

INSTRUCTIONAL TEAM:

Professor

Dr. Monique Ross	moross@fiu.edu	Office: ECS 389/Zoom Office hours: Upon request

COURSE DESCRIPTION: This course focuses on the design of large-scale computer programs. Included are the following topics: planning, design, implementation, validation, metrics, and the management of such software projects.

PREREQUISITES: CEN 4010

COURSE OBJECTIVES:

- Be **familiar** with planning and managing large scale software systems.
- **Demonstrate** mastery of techniques of analyzing and designing large scale software systems.
- **Demonstrate** mastery of software systems implementation.
- Be **familiar** with techniques used to test large scale software systems.
- Be **familiar** with the documents generate during the software development process.
- **Demonstrate** the ability to work effectively in a software development team.

Software engineering is the application of computer science. This course is designed to provide breadth not depth in a number of topics. These topics coalesce in a project in an attempt to create a realistic, practical application of the content presented in this course.

COURSE OUTCOMES:

- Develop and articulate an engineering design approach to solving a complex problem.
- Develop a software artifact with consideration for implications and customer needs.
- Communicate with your teammates and work effectively in a small team.

COURSE ACTIVITIES: Course activities will involve:

- Explore software engineering – via journal articles and discussion – and through reflection, writing and dialogue.
- Reflect on and discuss the implications of design decisions
- Develop a written argument for a design approach (e.g., the Project Paper described on subsequent pages).
- Developing an artifact under constraint, on-time, on-budget, with thoughtful and consistent results (Semester Project)
- Developing a workable, functioning product

READINGS:

We will use the following **written materials**:

We will also use chapters from various texts and several articles that will be made available on Canvas.
Other useful resources (Optional not required but useful)

- Ashmore, S., Runyan, K. (2015). Introduction to Agile Methods. ISBN# 0-321-92956-X

TOOLS: We will be utilizing three additional tools for the class.

1. Canvas for submitting work and supplemental reading assignments.

COURSE REQUIREMENTS AND GRADING:

Preparation

Weekly canvas quizzes

Participation

In-class Activities/Discussions

Project

Algorithmic Accountability Statement

Weekly Status Briefing

Project Paper – Design Document

Working Software Product

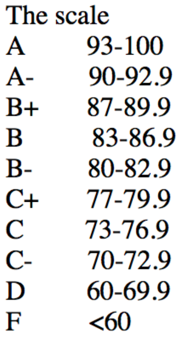
Final Presentation

It is my hope that by being transparent with my course design philosophies and practices, you as the student will, hopefully, understand that I have been thoughtful in my development of this course to ensure you have an effective learning experience. Given the nature of the discipline and the course I am most interested in your ability to transfer the knowledge you learn in this course into your future work (e.g., senior design project, internships, career). It is my belief that through engaging in this learning environment you will master the content, draw connections to personal interests, and push yourself to a higher level of thinking in computer science.

<u>Three assessment categories</u> will be used to determine your final grade for this course.

1. **15%** of your grade will be earned through **preparation –** <u>weekly quizzes</u> that cover in-class and reading material
2. **15%** of your grade will be earned by your **participation** as measured by <u>participation in in-class</u> activities and discussions
3. **70%** of your grade will be earned by participating in a **full project** . See pages 4-5 for a detailed description of the elements that must be part of the design document. Each section of the final paper will be assessed on the following three criteria: *clarity and meets the requirements* explicitly stated in this document. The project grade will consist of (1) algorithmic accountability statement (10%); (2) weekly status briefing (5%); (3) design document (25%); (4) working software product (15%); (5) final presentation (15%).

COURSE DELIVERY: Each week we will meet twice a week for 1 hour 15 minutes. One class per week will be dedicated to course content, one class per week will be dedicated to project-specific tasking or discussions (see Course Schedule).



The scale

A	93-100
A-	90-92.9
B+	87-89.9
B	83-86.9
B-	80-82.9
C+	77-79.9
C	73-76.9
C-	70-72.9
D	60-69.9
F	<60

LEARNING ENVIRONMENT:

The research is clear - long-term learning takes place only when accompanied by **deliberate, distributed practice**. To accomplish that end, this class is structured so that there is opportunity for reflection and iteration. Detailed feedback will be provided as you develop your ideas. My philosophy is that:

- Successful learning is fostered by clear objectives and expectations.
- Learning is a social activity, and you will be encouraged to "think together" with your classmates.
- Reading and writing are essential parts of the thinking process and you will be asked to use writing to DEVELOP your ideas (not just to document your "final" project).

<u>I expect that you will:</u>

1. Try to **<u>attend all classes</u>** and **<u>arrive on time.</u>** If you must miss a class, please let me know. I will record all classes, so material should be available online; however, you should make sure you are

communicating with your team as well. Communication is key.

2. Fully engage in all classes. Full engagement requires reading and reflecting on all assigned materials by the assigned time, actively participating in class discussions and activities, and completing quality work. Full engagement also means being attentive in class.
3. Develop your final project throughout the semester by developing drafts and continually reflecting on how course material relates to your project.
4. Follow scholastic conduct policy: http://undergrad.fiu.edu/academic-integrity/
   a. Pay specific attention to issues related to plagiarism.
5. Complete and submit a thoughtful course evaluation.

You can expect that I will:

1. Provide a supportive learning environment that fosters your success.
2. Create assignments and exercises that are meaningful to you.
3. Provide detailed, constructive formative feedback on your project drafts, upon request.
4. Honor and respect your interests.

From experience teaching this (or a similar) course, I have found that the most successful students:

1. Create early drafts that are thoughtful and complete.
2. Think deeply about their course project.
3. Are open to new ways of thinking about design and the target domain of their project.

ASSESSMENT

Preparation There will be weekly class quizzes that assess any material covered in class or assigned readings. They will be no longer than five questions long. All quizzes will be available in Canvas during the first 10 minutes of the class (Monday). If you are late, you will not be given additional time to complete the quiz. No make-ups. The intent is not to torture you but to a) ensure that you keep up with reading assignments, pay attention, and most importantly provide opportunities to practice memory retrieval. Studies show that multiple opportunities to retrieve new knowledge helps to create new neural pathways and helps you retain information longer (Lang 2016). Blame it on all that reading I do on how best to learn.

Participation Where appropriate there will be in-class activities (e.g., creating concept maps, minute-papers, muddiest-point, etc.) that require your preparation and participation…another opportunity to establish those neural pathways. You must attend class and be engaged with the material. Given the challenges related to COVID, I will NOT be taking attendance but that does not change my expectations that you will attend class.

Full Project

The principal assignment for the course is a **group project (and paper)**. The actual project/digital artifact (e.g., language, type, etc.) is left to the discretion of your team with an explicit explanation of decisions captured in your design document. If you wish to have my review the design document during the dates specified in the calendar (below) they must be handed in using Word or Pages format NOT .pdf (until the final draft). This allows me to put comments in-line in the document. The production of the final software engineering document will occur in a series of (highly recommended, optional) stages:

1. A first draft (optional) that begins to show the development of your ideas as well as an assessment of its social or societal need either with an actual customer or through scholarly literature. It will also include a schedule and backlog (requirements).
2. A second draft (optional) that further develops your ideas. Ideally, this draft will contain the design of your final project as well as incorporate feedback from previous draft.
3. A third and final draft (required) that further develops your ideas. Ideally, this draft will contain the verification plan of your final project as well as incorporate feedback from previous drafts.
4. A presentation of your project (required).

While draft one and two are optional, I strongly recommend that you take advantage of at least ONE of these optional drafts. Past experience suggests that in order to pass this class I should review and provide feedback for at least one draft before the final product.

The final paper will include the following areas. This is presented as an outline; however, you final product should read like a design document (see exemplars in Canvas) You **MUST** address all of the bullets below for full credit:

0. A title for your project that identifies your audience/customer/consumer, artifact and setting
   - a. i.e., “Binary Learning Application for Elementary Students”

1. Introduction
   - a. Description of the customer/setting for the project
     - i. Salient characteristics of the *customer* or sponsoring organization
     - ii. Description of the salient characteristics of the *application/tool*
       - 1. What is the background of the project idea? What is the problem?
       - 2. How is your application/tool different from what already exists?
       - 3. What are the implications of the tool you develop?
     - iii. Other important contextual issues especially any external constraints placed on the application/tool.
     - iv. Outline a proposed schedule for your project
       - 1. Show your User Story backlog
         - a. The user stories should follow the format: *As a ,<type of user>, I want <some goal>, so that <some reason>.*
         - b. It should also be followed by a set of Acceptance Criteria: *Verify that…*
       - 2. Describe the process used to determine user story allocation (e.g., MoSCoW, Story Points, Planning Poker, Level of Effort, or any other planning process.
       - 3. Determine the sprint period (1 – 3 weeks)
       - 4. Allocate the backlog to the Sprints
     - v. Describe your team and the proposed roles and responsibilities for your teammates
   - b. Your motivation for selecting this project //individual response
     - i. Your own expertise in the area
     - ii. How this might be useful to you in your career
2. Design
   - *a.* Define design //using scholarly literature to support your definition; following *APA citation*
   - b. Introduce your design process – the process invoked by your team
   - c. Demonstrate the design graphically
     - i. Use case Diagrams
     - ii. Activity Diagrams (if applicable)
     - iii. Class Diagrams
     - iv. Sequence diagrams
   - d. Rationale Management – be sure to provide a justification for design decisions made during the project (hint* - take notes as you are going through the process, journal even…this will help you with this portion of the project).
     - i. The issues that were addressed
     - ii. The alternatives that were considered
     - iii. The decisions that were made to resolve the issues
     - iv. The criteria used to guide decisions
     - v. The debate developers went through to reach a decision
3. Verification
   - a. Define verification //using scholarly literature to support your definition
   - b. Introduce your verification process – the process invoked by your team

   c. Describe/demonstrate your testcases – system level testcases ONLY (no unit testing necessary)
   d. Demonstrate (preferably a table) **traceability** between the testcases/plan and user story
4. Reflection
   a. In one or two paragraphs, discuss the lessons you learned during this project (individually) (e.g., teamwork, project selection, software life cycle, implications, schedule).
      i. Share the **actual** timeline for execution of the project. Compare the **proposed** schedule with the **actual** timeline. Describe and explain the deviation from the planned or proposed.

TEAM SELECTION: Complete the CATME poll to aid in team selection. The tool is available Week 1. You will receive an email when the survey is available. CATME will also be utilized for team evaluations. There will be three team evaluations. Completion of the CATME surveys are counted as part of your participation grade.

ACCOMODATIONS: If you are a person with special circumstances that you believe will affect your class performance (e.g., visual, hearing or learning disabilities or language differences) please let me know if I can make appropriate accommodations. The Disability Resource Services website is located at: http://studentaffairs.fiu.edu/get-support/disability-resource-center/

COURSE SCHEDULE for Fall 2019

Week	**Date**	**Class focus**	Readings *due*	**Deliverables *due***
1	Aug. 24	Course overview, introductions – assumptions and misconceptions		Familiarity Survey Introduction Quiz
	Aug. 26	What is software engineering? (concept mapping) Introduction to Software Engineering	Familiarize yourself with Draw.io; map function to use in class	
2	Aug. 31	Team selection/introductions Team guidelines	Assigned reading (see Canvas)	Quiz 1
	Sep. 2	Project Guidelines Project brainstorm session		
3	Sep. 7	HOLIDAY NO CLASS		
	Sep. 9	Algorithmic Accountability		Quiz 2
4	Sep. 14	Introduction to Software Engineering Waterfall vs. Agile Configuration Management	Assigned reading (see Canvas)	Quiz 3
	Sep. 16	Weekly Status training/exercise		
5	Sep. 21	Planning – user stories Sprint Planning • User Stories • Backlog Points/LOE	Assigned reading (see Canvas)	Quiz 4
	Sep. 23	Discussion/ Project Work		Algorithmic accountability statement
6	Sep. 28	Rationale Management Class – User stories	Assigned reading (see Canvas	Quiz 5
	Sep. 30	Discussion/ Project Work		

7	Oct. 5	Peer review training/exercise	Assigned reading (see Canvas)	Quiz 6
	Oct. 7	Discussion/ Project Work		
8	Oct. 12	UML	Assigned reading (see Canvas)	Quiz 7
	Oct. 14	Discussion/ Project Work		Weekly Status
9	Oct. 19	UML	Assigned reading (see Canvas)	
	Oct. 21	Discussion/Project Work		Weekly Status **Introduction Due (Sections 0-1)** ***OPTIONAL***
10	Oct. 26	Verification	Assigned reading (see Canvas)	Quiz 8
	Oct. 28	Discussion		Weekly Status
11	Nov. 2	Discussion/Project Work	Assigned reading (see Canvas) YouTube Videos	
	Nov. 4	Discussion/Project Work		**Design Due (Sections 0-2) OPTIONAL**
12	Nov. 9	Discussion/Project Work	Assigned reading (see Canvas)	Weekly Status
	Nov. 11	HOLIDAY NO CLASS		
13	Nov. 16	Discussion/Project Work		
	Nov. 18	Discussion/Project Work		Weekly Status
14	Nov. 23	Discussion/Project Work		
	Nov. 25	Discussion/Project Work		Weekly Status
15	Nov. 30	Discussion/Project Work		
	Dec. 2	Discussion/Project Work		**Verification Due (Sections 0-4) REQUIRED**
	FINALS – Presentations – FINALS (Dec. 7 – 12) TBD			
	YOU MADE IT!			

*Syllabus format and key points credited to Dr. Ruth Streveler, Purdue University

Create a network for peer review of deliverables; add points for participation in the peer review process; add rubrics