COMP 350: Software Engineering
Spring 2021

THIS IS A DRAFT SYLLABUS WHICH MAY BE UPDATED THROUGHOUT THE COURSE
Last updated: April 26, 2021

Instructor

Michael Soltys
michael.soltys@csuci.edu
http://www.msoltys.com
Sierra Hall 3327

Course Information

COMP 350, taught online, synchronously MW 11:00-1:15
Course web page: http://prof.msoltys.com/?page_id=5912
Office Hours: by appointment on Zoom.

Course Description

Concepts and techniques for systems engineering, requirements analysis, design, implementation and testing of large scale computer systems. Principles of software engineering for production of reliable, maintainable and portable software products. Emphasis on functional analysis and structured design techniques. Topics include unit, integration and systems testing, configuration management, and software quality assurance practices. Participation in group activities involving analysis, design and implementation of a software intensive system. Introduction to Computer Aided Software Engineering (CASE).

Prerequisite: COMP 232 and COMP 262

Student Learning Outcomes (SLOs)

1. Create effective documentation for a software project.
2. Organize and express ideas clearly and convincingly in oral and written forms.
3. Propose project plan.
4. Create a design document including requirements, specifications and division of duties among team members.
5. Implementation of design as a software product.
6. Recognize professional responsibilities and make informed judgments in computing practice and legal and ethical principles.

Textbook

- Required: *AWS Developing certification material*, to be provided electronically to the students at no cost.
- Recommended: *Clean Code: A Handbook of Agile Software Craftsmanship*, by Robert C. Martin.

Course Outline

Week	**Dates**	**Module**	**Project**	**SLOs**
1	Jan 25 – 29	1	course introduction	
2	Feb 1 – 5	2	organization of project using IAM on AWS	3,4
3	Feb 8 – 12	3	work on project design 1/2	3,4
4	Feb 15 – 19	4	work on project design 2/2	3,4
5	Feb 22 – 26	5	presentation of project design	2
6	Mar 1 – 5	6	work on project implementation	5
7	Mar 8 – 12	7	work on project implementation	5
8	Mar 15 – 19	∗∗∗∗∗∗∗ ***Spring Break*** ∗∗∗∗∗∗∗		
9	Mar 22 – 26	8	presentation of working prototype	2,5,6
10	Mar 29 – Apr 2	9	improvements following presentation	2,5
11	Apr 5 – 9	10	documentation and testing	1,2
12	Apr 12 – 16	11	integrating all components	
13	Apr 19 – 23	12	presentation of MVP	2
14	Apr 26 – 30	13	work & presentation of submittable	2
15	May 3 – 7	-	work & presentation of submittable	2
16	May 10 – 14	-	final presentation and delivery	2

Grading

The grade of the course consists of two components: 50% AWS Developing course, and 45% project. For the project, each student is required to do two presentations that describe their contribution; the grade will be based on the presentation and the contribution. There will also be a 5% "ethics assignment."

Grade determination

From	To	Letter Grade	From	To	Letter Grade
97	100	A +	77	79.99	C+
94	96.99	A	74	76.99	C
90	93.99	A-	70	73.99	C-
87	89.99	B+	67	69.99	D+
84	86.99	B	64	66.99	D
80	83.99	B-	60	63.99	D-
			0	59.99	F

Policies

1. **Academic Dishonesty:** By enrolling at CSU Channel Islands, students are responsible for upholding the University's policies and the Student Conduct Code. Academic integrity and scholarship are values of the institution that ensure respect for the academic reputation of the University, students, faculty, and staff. Cheating, plagiarism, unauthorized collaboration with another student, knowingly furnishing false information to the University, buying, selling or stealing any material for an examination, or substituting for another person may be considered violations of the Student Conduct Code (located at `http://www.csuci.edu/campuslife/student-conduct/academic-dishonesty.htm`). If a student is found responsible for committing an act of academic dishonesty in this course, the student may receive academic penalties including a failing grade on an assignment or in the course, and a disciplinary referral will be made and submitted to the Dean of Students office. For additional information, please see the faculty (located at `https://senate.csuci.edu/policies/2013-2014/sp-13-06-policy-on-academic-dishonesty-rev-oct2016.pdf`), also in the CI Catalog.

2. **Disability Statement:** If you are a student with a disability requesting reasonable accommodations in this course, please visit Disability Accommodations and Support Services (DASS) located on the second floor of Arroyo Hall, or call 805-437-3331. All requests for reasonable accommodations require registration with DASS in advance of need: `https://www.csuci.edu/dass/students/apply-for-services.htm`. Faculty, students and DASS will work together regarding classroom accommodations. You are encouraged to discuss approved accommodations with your faculty.

3. **Course Policies Subject to Change:** It is the student's responsibility to check the course's web page frequently to stay abreast of the course, and for corrections or updates to the syllabus. Any changes will be posted there.

Course Assessment

Computer Science Student Learning Outcome (SLO) "4." states:

> *Recognize professional responsibilities and make informed judgments in computing practice and legal and ethical principles.*

Here is the rubric for this outcome:

Performance Indicator	**Unsatisfactory**	**Developing**	**Satisfactory**	**Exemplary**
1. **Recognize an ethical dilemma:** can read a software engineering ethics related case study and recognize a dilemma.	no problem recognition	identifies a single dilemma without recognizing ramifications	identifies key ethical dilemma, but unable to reconcile opposing alternatives	clearly identifies and frames key ethical dilemmas, and grasps that there are opposing alternatives that must be recognized
2. **Information:** Can read a software engineering ethics related case study and understand the relevance of pertinent facts needed to evaluate the situation.	Ignored pertinent facts or used misinformation	Lists information without explaining its relevance and does not state assumptions	Lists information and explain its relevance, but does not explicitly state assumptions	Lists information, explains its relevance, and explicitly states assumptions with justifications. The student may bring in information from their own experience.
3. **Analysis:** Can critically analyze a software engineering ethics related case study.	No analysis	Rule driven without justification	Demonstrates awareness of multiple alternatives and makes some attempt to compare and contrast them	Provides a thorough analysis, cites analogous cases, and considers risk elements with respect to each alternative
4. **Perspective:** Can read a software engineering ethics related case study and recognize the multiple points of view.	Wandering focus indicating lack of perspective	Articulates single point of view	Acknowledges multiple points of view, perhaps articulating the case from multiple points of view	Provides a thorough analysis, cites analogous cases, and considers risk elements with respect to each alternative
4. **Resolution:** Can read a software engineering ethics related case study and recommend an appropriate resolution.	Not responsive to scenario	Cited rules as the resolution, even if used out of context	Resolution considers the potential risk to the public and/or safety, and other stakeholders	Considers potential risk and/or public safety, and proposes a creative middle ground (win-win)

The threshold will be 80%, that is, at least 80% of students must meet the “satisfactory” or “exemplary” level. All five rows will be measured by a case study undertaken by students.