School of Computing and Information Sciences

Course Title: Software Engineering I **Date:** May 7, 2020

Course Number: CEN 4010

Number of Credits: 3

Subject Area: Software Engineering	**Subject Area Coordinator:** Monique Ross **email:** moross@cs.fiu.edu
Catalog Description: Software Process Model, Software Analysis and Specification, Software Design, Software Testing	
Textbook: Ashmore, S., Runyan, K. (2015). Introduction to Agile Methods, ISBN#: 978-0-321-92956-3 Fowler, M. (2004). UML Distilled: A brief guide to the standard object modeling languge, 3rd edition	
References:	
Prerequisites Courses: CGS 3095 and COP 3337	
Corequisites Courses: None	

Type: Required for CS Major

Prerequisites Topics:

- Programming
- Data Structures
- Oral and written communication skills

Course Outcomes:

1. Be familiar with the Software Development Life Cycle and software process models
2. Master the techniques to gather and specify the requirements of a medium-size software system
3. Master the techniques to design and implement a medium-size software system using UML
4. Be familiar with software testing techniques
5. Be familiar with system walkthroughs
6. Be familiar with software documentation
7. Be familiar with working in a small software development team
8. Demonstrate the ability to communicate the details of the technical solution through verbal and written modes.

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Relationship between Course Outcomes and Program Outcomes

BS in CS: Program Outcomes	Course Outcomes
a) Demonstrate proficiency in the foundation areas of Computer Science including mathematics, discrete structures, logic and the theory of algorithms	
b) Demonstrate proficiency in various areas of Computer Science including data structures and algorithms, concepts of programming languages and computer systems.	3, 4
c) Demonstrate proficiency in problem solving and application of software engineering techniques	1, 2, 3, 4, 5, 6, 7
d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.	
e) Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.	
f) Demonstrate the ability to work cooperatively in teams.	2, 5, 7
g) Demonstrate effective communication skills.	2, 5, 6, 7, 8

Assessment Plan for the Course & how Data in the Course are used to assess Program Outcomes

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan: http://www.cis.fiu.edu/programs/undergrad/cs/assessment/

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Outline

Topic	Number of Lecture Hours	Outcome
• Introduction to Software Engineering o Concepts o Life Cycle Model o Products o Reviews o Development Team	6	1, 5, 6, 7
• Software Modeling o Concepts o Modeling with UML	8	1, 2, 6
• Requirement Gathering and Analysis o Concepts and Activities o Functional Requirement ▪ Scenarios and Use Cases o Non-functional requirements o Requirement Validation	12	1, 2, 5, 6, 7
• Software Design o System Design ▪ Design Goals ▪ Cohesion and Coupling ▪ Persistent Data ▪ Access Control o Object Design ▪ Object Interface ▪ Invariants ▪ Pre and post conditions	12	1, 3, 5, 6, 7
• Testing o Testing Concepts o Test Planning o System Level Testing o	4	4, 5

Learning Outcomes: (Familiarity→Usage→Assessment)

Software Processes:

1. Describe the relative advantages and disadvantages among several major process models (e.g., waterfall, iterative, and agile). [Familiarity]

2. Describe the different practices that are key components of various process models. [Familiarity]

3. Differentiate among the phases of software development. [Familiarity]

4. Describe how programming in the large differs from individual efforts with respect to understanding a large code base, code reading, understanding builds, and understanding context of changes. [Familiarity]

Requirements Engineering:

1. List the key components of a use case or similar description of some behavior that is required for a system. [Familiarity]

2. Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements. [Familiarity]

3. Interpret a given requirements model for a simple software system. [Familiarity]

4. List the key components of a data model (e.g., class diagrams or ER diagrams). [Familiarity]

5. Identify both functional and non-functional requirements in a given requirements specification for a software system. [Usage]

6. Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements. [Usage]

7.

Software Design:

1. Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design. [Usage]

2. Construct models of the design of a simple software system that are appropriate for the paradigm used to design it. [Usage]

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Course Outcomes Emphasized in Laboratory Projects / Assignments

Outcome	Number of Weeks
1. Software Requirement and Analysis Model Outcomes: 1,2,6,7	4
2. Software Design Document Outcomes: 1,3,6,7	4
3. Final Software Project Demonstration Outcomes: 1,2,3,4,5,6,7	4

Oral and Written Communication:

Written Reports		Oral Presentations	
Number Required	Approx. Number of pages for each	Number Required	Approx. Time for each
3 (Software Requirement, Design Document and Final System Document)	30	2	15 minutes per group (5 minutes per student)

Social and Ethical Implications of Computing Topics:

No significant coverage

Topic	Class time	Student Performance Measures

School of Computing and Information Sciences
CEN 4010
Software Engineering I

Approximate number of credit hours devoted to fundamental CS topics

Topic	Core Hours	Advanced Hours
Algorithms:		
Software Design:		2.0
Computer Organization and Architecture:		
Data Structures:		
Concepts of Programming Languages:		

Theoretical Contents

Topic	Class time
Invariants, pre and post conditions	1.0

Problem Analysis Experiences

Software requirement and analysis model

Solution Design Experiences

1. System Design using Architectural Patterns
2. Detailed Object design using Design Patterns

School of Computing and Information Sciences
CEN 4010
Software Engineering I

The Coverage of Knowledge Units within Computer Science Body of Knowledge[1]

Knowledge Unit	Topic	Lecture Hours
SE 1 SE/SWDesign 2	Fundamental Design concepts and principles, Software Architecture, Object-Oriented Design	12
SE 4 SE/SWProcess 2 SE/SWProcess 3 SE/SWProcess 4	Software Life-Cycle and Process Models	8
SE 5 SE/ReqEngr 1 SE/ReqEngr 2 SE/ReqEngr 3	Requirement Elicitation, Requirements Analysis Modeling Techniques, Functional and Nonfunctional requirements, Basic Concepts of Formal specification techniques	12
SE 6 SE/SWV&V 4	Validation Planning, Testing Fundamentals, Black-box and White-box testing, Unit, integration, validation and system testing, Object-Oriented Testing, Inspections	4
SE 8 SE/SWProjMgt 1	Team Management, Software measurement and scheduling techniques, Project management tools	2
SE 10	Pre and post assertions	1

[1]See https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf for a description of Computer Science Knowledge units