CS3343: SOFTWARE ENGINEERING PRACTICE

Effective Term
Semester A 2022/23

Part I Course Overview

Course Title
Software Engineering Practice

Subject Code
CS - Computer Science
Course Number
3343

Academic Unit
Computer Science (CS)

College/School
College of Engineering (EG)

Course Duration
One Semester

Credit Units
3

Level
B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
Nil

Precursors
CS3342 Software Design or equivalent

Equivalent Courses
Nil

Exclusive Courses
Nil

Part II Course Details

Abstract

This course aims to provide an opportunity for students, in small groups with real role-playing, to practise software engineering principles and techniques, through the development of a larger and really useful software system that meets stated requirements and quality standards, using those common practices used in the software industry

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Properly apply the principles and techniques of requirements specification and analysis, software design, implementation, testing, delivery, and maintenance to create a software application.		x	x	x
2	Present projects effectively.		x	x	
3	Write technical documentation in a clear and concise manner.		x	x	
4	Work effectively in a team environment.		x	x	

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Lecture presentations deliver the course materials (best practices of software engineering) to the students. Students are required to attend the lecture classes.	1, 2, 3, 4	

2	Tutorial and consultation	The tutorial and consultation sessions are used to review some major points of the course materials and hold group meetings for reviewing all possible artifacts generated by the project groups. Students can also raise and discuss issues related to the project.	1, 2, 3, 4	
3	Group project	Students will have to work in small groups to work on a realistic software development project to create a software application with technical documentation, in order to gain experience in applying the principles and techniques of software engineering, and to practise project management skills in a team environment. Each group is expected to define clearly the roles of individual members. Every member will do an oral presentation on the work accomplished, and, in particular, on the individual contribution to the group project. The group report records their project progress, the problems they encountered, and how they solved them, and their personal experience of playing the role in the project.	1, 2, 3, 4	

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Hands-on practical tests	1, 3	40	
2	Group project	1, 2, 3, 4	30	

Continuous Assessment (%)

70

Examination (%)

30

Examination Duration (Hours)

2

Additional Information for ATs

For a student to pass the course, at least 40% of the maximum mark for the continuous assessment and 30% of the maximum mark for the examination must be obtained.

Assessment Rubrics (AR)

Assessment Task

Hands-on practical tests

Criterion

1.1 CAPACITY for SELF-DIRECTED LEARNING to understand the tools and practices of software development.
1.2 ABILITY to EXPLAIN AND APPLY software testing techniques.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

Group project

Criterion

2.1 CAPACITY for SELF-DIRECTED LEARNING to apply the learnt practices to real problems, produce an application and write technical documents.
2.2 ABILITY to EXPLAIN AND DEMONSTRATE IN DETAIL about project management in a team environment.
2.3 ABILITY to APPLY the software development and testing procedures to produce a quality software system within a team.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
Examination

Criterion
3.1 CAPACITY for SELF-DIRECTED LEARNING to understand the tools and practices of software development.
3.2 ABILITY to EXPLAIN AND APPLY software development and testing procedures to produce high quality software systems.

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Part III Other Information

Keyword Syllabus
Software project management. Requirements elicitation and specification. Design. Implementation. Testing. Documentation. Presentation.

Syllabus

- Software project management
  Project planning and scheduling. Team organisations and role playing. Version control and configuration management. Documentation.
- Requirement elicitation and specification
  User requirements specification. Prototyping.
- Design
  Input design. Output design. User interface design. Designing for Change. CASE tools.
- System implementation and testing
  Integrated software engineering environments (IDE). Programming standards. Software testing strategies. Software Debugging. Refactoring. CASE tools.

Reading List

Compulsory Readings

	Title
1	Nil

Additional Readings

	Title
1	Sommerville I. (2012) Software Engineering. Addison Wesley, 10th edition.
2	Larman C. (2005) Applying UML and Patterns: Introduction to OOA/D and Iterative Development. Pearson Education, Prentice Hall, 3rd edition.
3	Martin R C, and Martin M. (2006) Agile Principles, Patterns, and Practices in C#. Prentice Hall.
4	Hughes B. and Cotterell M. (2005) Software Project Management. McGraw-Hill, 4th edition.
5	Christensen M. and Thayer R.H. (2002) A Manager's Guide to Software Engineering' s Best Practice. Wiley–IEEE Computer Society.