UNIVERSITY OF GUAM
UNIBETSEDÅT GUAHAN

COLLEGE OF NATURAL & APPLIED SCIENCES
Division of Mathematics & Computer Sciences

CS 477 Software Engineering
SYLLABUS

CLASS MEETING TIMES

Semester: Fall 2024
Section: CS477-01
Course Delivery Mode: Face-to-Face
Class Hours: 08:00-09:20 p.m. Tuesday & Thursday.
Classroom: Warehouse B Room 3

INSTRUCTOR

Name: Dr.Byoungyong Lee
Office: Warehouse B Room 4
Phone: 671-735-2831
E-mail: leeby@triton.uog.edu
Office Hours: Mon/Wed: 08:30 am – 09:30 am / 12:30 pm – 01:00 pm
Tue/Thu: 11:00 am – 12:00 pm / 03:30 pm – 04:00 pm

COURSE DESCRIPTION

This course presents software engineering techniques and examines the software life-cycle, including software specification, design, implementation, testing and maintenance. The course evaluates past and current trends in software development practices. The course gives an overview and discusses methods and techniques used in agile software processes, contrasts agile approaches with traditional software development methods.

PREREQUISITES

CS373 Data Structures and Algorithms and CS377 Database Design and Implementation

SKILLS & BACKGROUND REQUIRED OR EXPECTED

No other than skills and background obtained in the prerequisites.
Skills in Web design, utilizing cloud, and any other software technologies may be useful.

REQUIRED TEXTBOOK, EQUIPMENT, AND/OR READINGS

Roger S. Pressman and Bruce Maxim, Software Engineering: A Practitioner's Approach, (9th ed) McGraw-Hill

Other possible textbooks/references

Ian Sommerville, Software Engineering (10th Edition), Pearson;

Gene Kin, Patrick Debois, John Willis, Jez Humble, and John Allspaw, The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, IT Revolution Press

Other recommendations

Students are encouraged to subscribe to online trade magazines such as Application Development Trends, Application developer Magazine, Information Wee, Java magazine, etc.

RATIONALE FOR COURSE

Successful software development depends on an in-depth understanding of how the phases and supporting activities of the software development life cycle work together. This course presents modern software engineering techniques and examines the software life-cycle, including software specification, design, implementation, testing and maintenance.

COURSE FORMAT

The course consists of lecture, group discussions and presentations for project team, and hands-on exercises.

GRADING SYSTEM/EVALUATION METHODOLOGIES

Course Requirements	Percent (%)
Attendance	10%
Homework	15%
Presentation	15%
Projects	15%
Midterm Exam	20%
Final Exam	25%
Total	100%

Letter grades will be assigned per the UOG Catalog:

A	90 – 100%
B	80 – 89%
C	70 – 79%
D	60 – 69%
F	0 – 59%

Homework

Reading assignment and/or problem sets will be given from the textbook and other instructional materials.

Presentation

Students form groups of 3 to 4 members and present group assignments.

Projects

1~2 projects will be given, and projects will be a team or individual projects.

Midterm Exam: Comprehensive

Final Exam: Comprehensive

COURSE TOPIC/EXAM SCHEDULE

Week	Topic	
1-2	• Chapter 1: Software and Software Engineering • Chapter 2: Process Model	
3 - 4	• Chapter 3: Agility and Process • Chapter 4 : Recommended Process Model • Chapter 5: Human Aspects of Software Engineering • Chapter 6: Principles That Guide Practice	
5 - 6	• Chapter 7: Understanding Requirements • Chapter 8: Requirements Modeling	
7 - 8	• Chapter 9: Design Concepts • Chapter 10: Architectural Design	**Midterm Exam**
9	Fall Break	Fall Break
10 - 11	• Chapter 11; Component Level Design • Chapter 12: User Experience Design	
12 - 13	• Chapter 13: Design for Mobility • Chapter 14: Pattern-Based Design • Chapter 15: Quality Concepts	
14 - 15	• Chapter 16: Reviews-A Recommended • Chapter 17: Software Quality Assurance • Chapter 18: Software Security Engineering	
16 -17	• Chapter 19: Software Testing – Component Level • Chapter 20: Software Testing - Integration Level	
18		**Final Exam**

Student Learning Outcomes (SLOs) : This course covers the following ACM/IEEE CC2013 Body of Knowledge student learning outcomes.

- **SE/** Software Process
- **SE/** Software Project Management
- **SE/** Requirements Engineering
- **SE/** Software Design
- **SE/** Software Construction
- **SE/** Software Validation and Verification
- **SE/** Software Evolution
- **SE/** Formal Methods

Maps to Program Learning Outcomes and Institutional Learning Outcomes

CS 477 Student Learning Outcomes (SLO)	Program Learning Outcomes (PLO)*	Institutional Learning Outcomes (ILO)*	Activities/Assessments
4, 5, 6, 12, 17, 18-20	1	1	Homework, project, exam
7, 11, 18-20, 23	2	1	Homework, project, exam
2, 10, 14, 27	3	3	Homework, project, exam
3, 8a, 8d, 9, 15b,	5	4	Homework, project, exam
1, 8b, 8c, 8e, 23	6	1	Homework, project, exam

CS 477 Student Learning Outcomes (SLOs) : Upon the completion of the course successfully, Students will be able to

SLO-1. Apply
  a. a conflict resolution strategy in a team setting. [Usage]
  b. formal specification and analysis techniques to software designs and programs with low complexity. [Usage]
  c. simple examples of patterns in a software design. [Usage]

SLO-2. Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation. [Familiarity]

SLO-3. Assess and provide feedback to teams and individuals on their performance in a team setting. [Usage]

SLO-4. Compare and contrast integration strategies including top-down, bottom-up, and sandwich integration. [Familiarity]

SLO-5. Compare several common process models with respect to their value for development of particular classes of software systems taking into account issues such as requirement stability, size, and non-functional characteristics. [Usage]

SLO-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]

SLO-7. Construct models of the design of a simple software system that are appropriate for the

paradigm used to design it. [Usage]

SLO-8. Create

a. a team by identifying appropriate roles and assigning roles to team members. [Usage]
b. a set of tests for a medium-size code segment and document them. [Usage]
c. program assertions for a variety of behaviors ranging from simple through complex. [Usage]
d. an agenda for a team meeting. [Usage]
e. appropriate models for the structure and behavior of software products from their requirements specifications. [Usage]

SLO-9. Demonstrate through involvement in a team project the central elements of team building and team management. [Usage]

SLO-10. Describe

a. a form of refactoring and discuss when it may be applicable. [Familiarity]
b. different categories of risk in software systems. [Familiarity]
c. how software can interact with and participate in various systems including information management, embedded, process control, and communications systems. [Familiarity]
d. secure coding and defensive coding practices. [Familiarity]
e. techniques for identifying significant test cases for integration, regression and system testing. [Familiarity]
f. techniques, coding idioms and mechanisms for implementing designs to achieve desired properties such as reliability, efficiency, and robustness. [Familiarity]
g. the fundamental challenges of and common techniques used for requirements elicitation. [Familiarity]
h. the impact of risk in a software development lifecycle. [Familiarity]
i. the process of analyzing and implementing changes to an existing code base. [Familiarity]
j. the relative advantages and disadvantages among several major process models (e.g., waterfall, iterative, and agile). [Familiarity]
k. the role formal specification and analysis techniques can play in the development of complex software [Familiarity]
l. the role that tools can play in the validation of software. [Familiarity]
m. For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system. [Familiarity]
n. the sources, hazards, and potential benefits of team conflict. [Usage]
o. the phases of software development. [Familiarity]

SLO-11. Design a contract for a typical small software component for use in a given system. [Usage]

SLO-12. Discuss

a. the advantages and disadvantages of different types of software reuse. [Familiarity]
b. the challenges of evolving systems in a changing environment. [Familiarity]

SLO-13. Estimate the impact of a change request to an existing product of medium size. [Usage]

SLO-14. Explain
- a. the concept of a software lifecycle and provide an example, illustrating its phases including the deliverables that are produced. [Familiarity]
- b. the potential benefits and drawbacks of using formal specification languages. [Familiarity]
- c. the relationships between the requirements for a software product and its design, using appropriate models. [Assessment]

SLO-15. Identify
- a. Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server. [Familiarity]
- b. necessary roles in a software development team. [Usage]
- c. both functional and non-functional requirements in a given requirements specification for a software system. [Usage]
- d. the principal issues associated with software evolution and explain their impact on the software lifecycle. [Familiarity]

SLO-16. Interpret a given requirements model for a simple software system. [Familiarity]

SLO-17. Investigate the impact of software architectures selection on the design of a simple system. [Assessment]

SLO-18. Select an appropriate design paradigm for a simple system suitable for a given scenario,. [Usage]

SLO-19. Select and use a defined coding standard in a small software project. [Usage]

SLO-20. Select suitable components for use in the design of a software product. [Usage]

SLO-21. Track the progress of some stage in a project using appropriate project metrics. [Usage]

SLO-22. Undertake, as part of a team activity, an inspection of a medium-size code segment. [Usage]

SLO-23. Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design. [Usage]

SLO-24. Use an ad hoc method to estimate software development effort (e.g., time) and compare to actual effort required. [Usage]

SLO-25. Use refactoring in the process of modifying a software component. [Usage]

SLO-26. Using a common formal specification language, formulate the specification of a simple software system and derive examples of test cases from the specification. [Usage]

SLO-27. Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system. [Familiarity]

CS Program Learning Outcomes (PLOs)

- *PLO-1.* Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- *PLO-2.* Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
- *PLO-3.* Communicate effectively in a variety of professional contexts.
- *PLO-4.* Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- *PLO-5.* Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
- *PLO-6.* Apply computer science theory and software development fundamentals to produce computing-based solutions.

UOG Institutional Student Learning Outcomes (ILOs)

- *ILO-1.* Critical thinking and problem solving
- *ILO-2.* Mastery of quantitative analysis
- *ILO-3.* Effective oral and written communication
- *ILO-4.* Understanding and appreciation of culturally diverse people, ideas and values a democratic context
- *ILO-5.* Responsible use of knowledge, natural resources, and technology
- *ILO-6.* An appreciation of the arts and sciences
- *ILO-7.* An interest in personal development and lifelong learning

COURSE POLICIES

Assignment

- Late labs and homework assignments with receive late penalties.

Midterm and Final Exam

- There are no make-up exams, unless with the consent of the instructor.

Attendance : Class attendance is mandatory.

Regular and punctual class attendance is expected of all students. Student must accept the consequences of failure to attend. Instructor will drop a student from the course for excessive absences. "Excessive absences" means failure to attend 70% of scheduled class meetings. A student missed more than 30% of scheduled classes will be dropped from the course and will receive a failing grade F. A student so dropped may appeal through the college's Due Process.

Note: Student who arrives after the instructor starts a class will be considered as tardy. 3 tardy will be counted as 1 unexcused absence.

A course for which a student registers and does not attend and is not officially dropped will be recorded as an "F" grade on the student's record. All students (including those who enroll in classes late) are responsible for the work covered and assigned from the first meeting of a class.

ACADEMIC DISHONESTY

Academic Integrity is about performing in your role as a student in ways that are honest, trustworthy, respectful, responsible, and fair (see www.academicintegrity.org for more information). As a student, you will complete your academic assignments in the manner expected by the instructor. Academic dishonesty, including but not limited to cheating and plagiarism may result in suspension or expulsion from the University. Refer to the UOG Student Handbook and Code of Conduct for more information.

Professional and ethical conduct is expected at all times. Unethical conduct includes any form of cheating, including plagiarism. The term "**cheating**" includes, but is not limited to: (1) use of any unauthorized assistance in taking quizzes, tests, or examinations, e.g., looking at other students' answers, using crib notes (including electronic), getting information from another person via any kind of communication; (2) dependence upon the aid of sources beyond those authorized by the instructor in writing papers, preparing reports, solving problems, or carrying out other assignments; or (3) the acquisition, without permission, of tests or other academic material belonging to a member of the University faculty or staff. If you need to use an electronic translator, you must discuss this with me in advance. All assignments and tests must be your own work. Answers you write on the tests must come only from in your head or the information supplied in the test papers; anything else is cheating. Any evidence of cheating will result in a "0" for that assignments and/or exam or possibly an "F" for the entire course – final decision to be determined by me, the course instructor.

FAMILY EDUCATIONAL RIGHTS AND PRIVACY ACT (FERPA)

The Family Educational Rights and Privacy Act (FERPA) affords students certain rights with respect to their education records. These rights for students, parents and school officials can be viewed at: **http://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html**

UOG DISABILITIES POLICY

In accordance with the Americans with Disabilities Act (ADA) of 1990 and the Rehabilitation Act of 1973, the University of Guam does not discriminate against students and applicants on the basis of disability in the administration of its educational and other programs. The University offers reasonable accommodations for a student or applicant who is otherwise qualified, if the accommodation is reasonable, effective and will not alter a fundamental aspect of the University's program nor will otherwise impose an undue hardship on the University, and/or there are not equivalent alternatives. Students are expected to make timely requests for accommodation, using the procedure below.

ADA Accommodation Services

For individuals covered under the ADA (Americans with Disabilities Act), if you are a student with a disability requiring academic accommodation(s), please contact the Disability Support Services Office to discuss your confidential request. A Faculty Notification letter from the Disability Support Services counselor will be provided to me. To register for academic accommodations, please contact or visit Sallie S. Sablan, DSS counselor in the School of Education, office 110, disabilitysupport@triton.uog.edu or telephone/TDD 671-735-2460.

TOBACCO-FREE/SMOKE-FREE/VAPING FREECAMPUS

UOG is a tobacco-free/smoke-free, vaping/e-cigarette free campus. Thank you for not using tobacco products on campus, and for helping make UOG a healthy learning and living environment. For more information visit: http://www.uog.edu/smoke-free-uog