CS1530: Software Engineering

FALL, 2024

INSTRUCTOR: Lee GU
OFFICE: N525
EMAIL: lee.gu@scupi.cn
OFFICE HOURS: Tuesday 10:00-13:30
LECTURES: Tuesday 13:50-16:25 & Wednesday 8:15-11:00
RECITATION: N211 (Tuesday) & 3-105 (Wednesday)
TEXTBOOK: Ian Sommerville, "Software Engineering", Pearson Publish.
TEACHING ASSISTANT:
Gao Ruihan; Guo Zhengmin

PREREQUISITE:
Comprehensive Knowledge on computer program language, data structure, computer architecture, and database.

DESCRIPTION:
Successful software development depends on an in-depth understanding of how the phases and supporting activities of the software development life cycle work together. Each phase of the life cycle contributes to a reliable, maintainable product that satisfies user requirements. The application of good engineering practices throughout the cycle dramatically improves the likelihood of delivering a quality software project on time, in scope and within budget. While there are many rigorous methodologies, in fact most approaches and tools have a mixture of strengths and weaknesses. Many modeling approaches focus on describing software designs, rather than solving business problems.
This course covers basic software engineering methodologies, techniques, and tools for planning, capturing requirements, designing, implementing, testing, and maintaining large-scale software systems.

COURSE OBJECTIVES:

- **Describe and compare various software development methods and understand the context in which each approach might be applicable.**
  - **How to elicit requirements from a client and specify them**
  - **Design in the large, including principled choice of a software architecture, the use of modules and interfaces to enable separate development, and design patterns.**
  - **Understanding good coding practices, including documentation, contracts, regression tests and daily builds.**
  - **Various quality assurance techniques, including unit testing, functional testing.**
- **Develop students' critical skills to distinguish sound development practices from ad hoc practices, judge which technique would be most appropriate for solving large-scale software problems.**
- **Expand students' familiarity with mainstream languages used to model and analyze object designs (e.g., UML).**

LEARNING OUTCOMES FOR THIS COURSE:

1) Understand basic knowledge on the scope of software engineering.

2) Build skills on working with version control, configuration management, unit/regression testing, issue tracking, and debugging tools; creating a project plan; creating and analyzing design models;.

3) Familiar to modern approach to cope with large scale software system development in teamwork.

GRADE DETERMINATION:

EXAMS: 40%

PROJECTS: 50%

ATTENDANCE and DISCUSSION: 10%

MATERIAL COVERED: The sequence of the sections covered in this class is:

Week	Contents	Descriptions
1 (09/03-04)	1.1 – 1.3	Introduction to software engineering
2 (09/10-11)	1.4 – 1.5	Ethics in SE, Case study and project assignment
3 (09/17-18)	2.1 – 2.2	Software process 1: Software process models & Process activities
4 (09/24-25)	2.3 – 2.5	Software process2: Coping with change & The Rational Unified Process
5 (10/02-03)		holiday
6 (10/08-09)	3.1 – 3.3	Agile Software Development
7 (10/15-16)	3.4-3.5	Extreme programming, Agile project management, Scaling agile methods
8 (10/22-23)	4.1 – 4.3	The software requirements document, Requirements specification
9 (10/29-30)	TA special	UML: basic tech & understanding
10(11/05-06)	4.4 – 4.5	Requirements elicitation and analysis, validation, management
11 (11/12-13)	4.6 & discussion	Requirement management & group discussion on requirement report
12 (11/19-20)	5.1 – 5.3	System Modeling
13 (11/26-27)	6.1 - 6.3	Architectural design decisions Architectural views and patterns
14 (12/03-04)	7 & discussion	Design and Implementation & group discussion on design report
15 (12/10-11)	8.1-8.3	Software Testing & group discussion on testing report
16 (12/17-18)	TA special	Project final rebuttal, presentation
17 (12/25)		Final Exam Week