SOFTWARE ENGINEERING

COURSE OUTLINE

TUẦN 1

Objectives

- Introduction to the field of software engineering.
- Provide undergraduate students with techniques, methods, processes and project management for the development of software-intensive systems.
- Activities for software development: requirements analysis, architectural & detailed design with patterns, coding, and testing.
- UML modeling language

Student learning outcomes

- (1) Understand the needs of professional software development;
- (2) Carry out software development activities such as requirements, design, coding, etc;
- (3) Use the UML language effectively in software development;
- and (4) Apply the most-frequently used design patterns and architectural patterns to the design phase of the development of software systems.

Textbook/reference book

- [1] Ian Sommerville (2010), Software Engineering (9th ed.), ISBN 978-0-137-03515-1, Addison Wesley
  - http://www.cs.st-andrews.ac.uk/~ifs/Books/SE9/
- [2] G. Booch, J. Rumbaugh, I. Jacobson (1998), The Unified Modeling Language User Guide, Addison-Wesley.
- [3] E.J. Braude (2001), Software Engineering: An Object-Oriented Perspective, ISBN 978-0-471-32208-5, John Wiley.
- [4] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, ISBN 978-0201633610, AddisonWesley Professional (Nov. 10, 1994)

Evaluation

- Lab + project: 20%
- In-class activities: 10%
- Mid-term exam: 20% (multiple choice)
- Final exam: 50% (multiple choice + writing)

Learning strategy

- Reading materials before the lectures
- Attending lectures => Activities on the lectures: 10%
- Preparing homework:
  - They are just summaries about the lectures —> exams
- Attending and doing labs: 5% of grade
- Merging lab works into SRS and SDD: 15% of grade
- Attending midterm and final exams: 20% and 50% of grade

LAB + Project

- Lab contents: group work on some problem given on the Lab
- UML tool: PatternWeaver
- Project: TBD
  - Could possible be train, movies, travel; ticket class: no-return, flexible, … with some option or refund policies
  - Mutual-exclusion, seat reservation
  - Architecture: connect to some payment system + Design pattern
- SRS & SDD: merge works from previous Labs into one document (group)

Tentative schedule

W/S	Topic	Reading	Lab
1	Introduction	Ch1[1], Ch0[3], IEEE	NO
2	Software process	Ch2[1], Ch1[3]	Lab intro
3-4	Requirement engineering	Ch4[1], Ch3-4[3], [2]	Proj. sched. + UML
5-6	System modeling	Ch5[1]	UML
7	Review / free talk		SRS practice
8	Midterm (in-class)		SRS deadline
9	Software architecture	Ch6[1], Ch5[3]	UML
10-12	Detail design	Ch7[1], Ch6[3], [2]	UML
13	Software implementation	Ch7-9[3]	SDD practice
14	Software testing & evolution	Ch8-10[1], Ch8-9[3]	SDD practice
15	Review / free talk		SDD deadline
16-19	Final exam (univ. schedule)		

Contact

- Lecturers:
  - Bui Hoai Thang (bhthang@hcmut.edu.vn)
  - Le Lam Son (lamson@hcmut.edu.vn)
  - Nguyen Cao Tri (caotri@hcmut.edu.vn)
  - Nguyen Trung Truc (truc@cse.hcmut.edu.vn)
  - Email subject: [SE] …
- Course website:
  - https://elearning.cse.hcmut.edu.vn/

Reference sources of the slides

- Slides in this course are adapted mainly from [1]. Some slides are adapted from [3].
- Slides of chapter “Software Implementation” are adapted from [3].