SOFTWARE ENGINEERING

CO3001

COURSE OUTLINE

Truong Tuan Anh

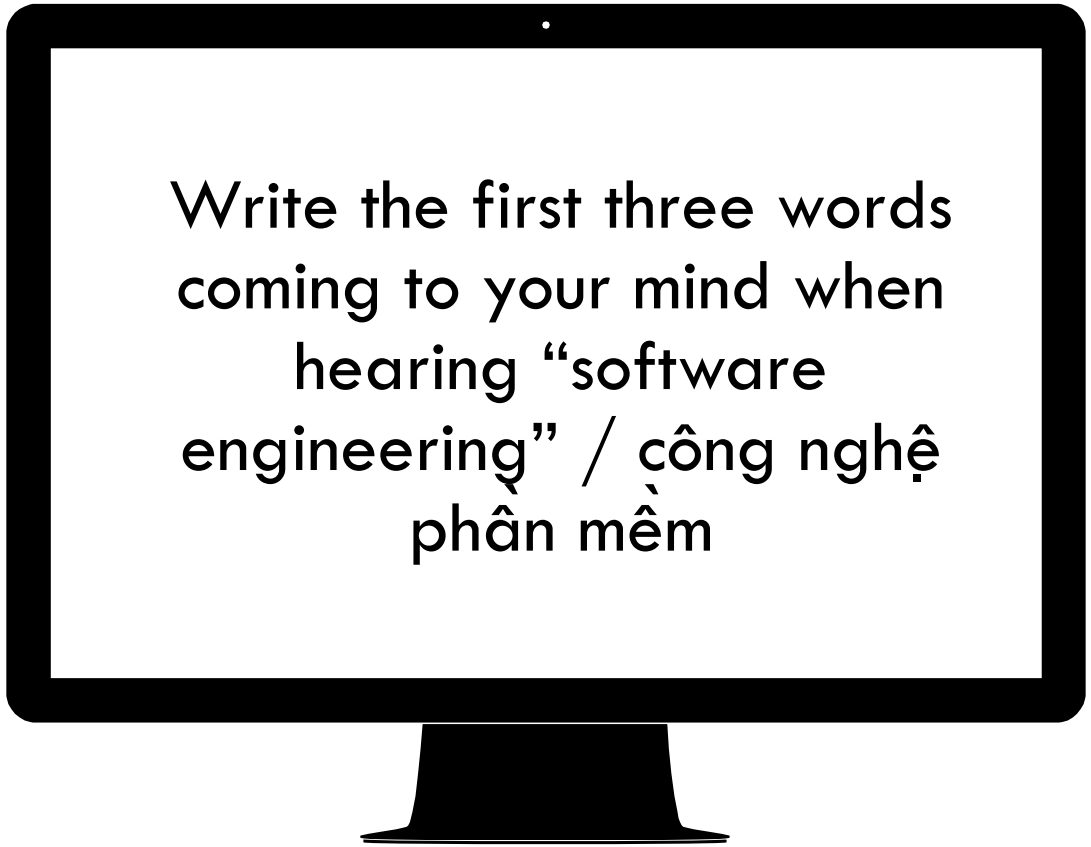
WEEK 1

Adapted from https://iansommerville.com/software-engineering-book/slides/

AIMS

- ✓ The goal of this course is to provide undergraduate students with
  - Knowledge (concepts, terms, processes, models)
  - Skill (methods, techniques)
- ✓ for requirement, analysis, design, implementation and testing of software-intensive systems.

Menti.com


Write the first three words coming to your mind when hearing “software engineering” / công nghệ phần mềm

OUTLINE

- ✓ An introductory course to the field of software engineering.
- ✓ The goal is to provide techniques, methods and processes for the development of software-intensive systems.
- ✓ Help getting familiar with software engineering activities: requirements elicitation, software specification, architectural & detailed design using design patterns.
- ✓ Also cover software implementation and software testing
- ✓ Use extensively the UML modeling language

STUDENT LEARNING OUTCOMES

Knowledge:

- ✓ L.O.1. Understand that software systems need to be developed methodologically and professionally;
- ✓ L.O.2. Elicit requirements & perform architectural design;

Competence:

- ✓ L.O.3. Cary out detailed design, coding, testing;
- ✓ L.O.4. Use the UML language effectively in software development.

STUDENT LEARNING OUTCOMES

No.	Course learning outcomes
L.O.1	Understand that software systems need to be developed methodologically and professionally;
	L.O.1.1 Understand principles and concepts of software engineering L.O.1.2 Understand methods and techniques of software engineering
L.O.2	Elicit requirements & perform architectural design
	L.O.2.1 Requirements elicitation L.O.2.2 Architectural design
L.O.3	Cary out detailed design, coding, testing
	L.O.3.1 Detailed design L.O.3.2 Coding L.O.3.3 Testing
L.O.4	Use the UML language effectively in software development
	L.O.4.1 UML use-case diagram L.O.4.2 UML sequence diagram L.O.4.3 UML class diagram L.O.4.4 UML activity diagram (or UML state-chart diagram)

TEXTBOOK/REFERENCE BOOK

- ✓ [1] Ian Sommerville (2015), Software Engineering (10th ed.), ISBN 978-0133943030, Pearson
- ✓ [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)
- ✓ [5] State-of-the-art articles on Software Engineering

TEACHING ACTIVITIES

- ✓ Read materials before the lectures
- ✓ Attend lectures
- ✓ Student presentation
- ✓ Group assignment
- ✓ Final exam

EVALUATION

- ✓ In-class/online activities/quizzes: 10%
- ✓ Student presentation: 10%
- ✓ Group-based project: 30%
- ✓ Final writing exam: 50%

QUIZZES

- ✓ Online, every week. Two categories:
  - Before the lecture:
    - Read slides, prepare yourself and take the quizzes before the lecture.
    - The questions are simple, just review the content of the coming lecture.
  - After the lecture (named "... - advanced"):
    - Summarize the content of the lecture, give to cases for you to apply the knowledge of the lecture.
    - The questions are some more advanced. Take your time to reflect the lecture before taking the quizzes.
- ✓ Duration limit of 10', valid in a few days.
  - You can take a quiz twice to get the highest score.
- ✓ Please check the video course website regularly:
  - http://e-learning.hcmut.edu.vn/user/index.php?id=72288

PROJECT

- ✓ Project:
- ✓ Group project
  - Team work
- ✓ Deliveries:
  - #1: Requirement elicitation: Functional/non-functional requirement, use-case and Wireframes
  - #2: Requirement analysis: Sequence, activity or state-chart diagrams
  - #3: Architectural design: Overall architecture, class diagram, implementation diagram
  - #4: First Demo
  - #5: Final Demo

STUDENT PRESENTATION

- ✓ Final:
  - Overview of project outcomes
  - Reasons behind technical decisions
  - Identified risk and project constraints

TENTATIVE SCHEDULE

Wk	Topic	Reading	Quiz	Project Milestone
1	Ch1. Introduction	Ch1[1], Ch0[3], IEEE	Quiz#1	
2	Ch2. Software process	Ch2[1], Ch1[3]	Quiz#2	Proj Introduction
3	Ch3. Req. engineering	Ch4[1], Ch3-4[3], [2]	Quiz#3	
4	Ch4. Req. engineering (cont.)	Ch4[1], Ch3-4[3], [2]		Proj#1
5	Review Project #1			
6	Ch6. System modeling	Ch5[1], [2]	Quiz#5	Proj#2
7	Review Project #2			
	Midterm break			
8	Ch7. Architecture design	Ch6[1], Ch5[3]	Quiz#6	
9	Ch8. Design and Implementation	Ch7[1], Ch6[3], [2]	Quiz#7	Proj#3
10	Ch9. Quality assurance	Ch7[1], Ch6[3], [2]	Quiz#8	
11	Ch10. Agile Software Development	Ch3[1]	Quiz#9	Proj#4
12	Review Project #3, #4			
13	Ch11. Continuous Integration/ Deployment			
14	Student presentation			Proj#5
15	Ch12. Advanced topics in SE	SE & Course Review		

CONTACT

- ✓ Lecturers team
  - Trương Tuấn Anh (anhtt@hcmut.edu.vn)
  - Mai Đức Trung (mdtrung@hcmut.edu.vn)
  - Trần Trương Tuấn Phát (phatttt@hcmut.edu.vn)
- ✓ Course website:
  - http://lms.hcmut.edu.vn

REFERENCE SOURCES OF THE SLIDES

- ✓ Slides in this course are adapted mainly from Sommerville 2015 [1]. Some slides are adapted from Braude 2001 [2].
- ✓ Slides of chapter "7.3. More on Implementation" are adapted from Braude 2001 [2].

[1] Ian Sommerville (2015), Software Engineering (10th ed.), ISBN 978-0133943030, Pearson
https://iansommerville.com/software-engineering-book/slides
[2] E.J. Braude (2001), Software Engineering: An Object-Oriented Perspective, ISBN 978-0-471-32208-5, John Wiley.