

# SOFTWARE ENGINEERING

## 1. Thông tin về học phần (General Information)

**Tên học phần (Course name):** Software Engineering

**Mã học phần (Course code):** INT1340

**Số tín chỉ (Number of credits):** 3

**Loại học phần (Course type):** Compulsory

**Học phần tiên quyết (Prerequisites):**

Introduction to computing and programming

**Học phần trước (Previous courses):**

**Học phần song hành (Parallel courses):**

**Các yêu cầu đối với học phần (Course requirements):**

- Lecture room: Projector, microphone and speaker, air conditioner.
- Laboratory:

**Giờ tín chỉ đối với các hoạt động (Teaching and Learning hours):**

- Lectures (lí thuyết): 36h
- Exercises (bài tập): 0h
- Projects (bài tập lớn): 08h
- Lab (thực hành): 0h
- Individual reading (tự đọc): 01h

**Địa chỉ Khoa/Bộ môn phụ trách học phần (Address of the Faculty/Department in charge of the course):**

- Address: Faculty of Information Technology 1 - Posts and Telecommunications Institute of Technology, Km10, Nguyen Trai Street, Ha Dong District, Hanoi.
- Phone number: (024) 33510432

## 2. Mục tiêu học phần (Objectives)

**Về kiến thức (Knowledge):**

The aim of this course is to provide learners with the understanding and skills for understanding and applying the techniques and method along the software development process, including software process, software life-cycle models, object-oriented software analysis and design tools. Learners will put this into practice by developing software from scratch.

**Kỹ năng (Skills):**

On successful completion of this course, a learner will be able to:

- use tools in software engineering
- apply the object-oriented analysis, design, implementation and testing technique into the development of a software

**Thái độ, Chuyên cần (Attitude):**

Learners are required to attend the classes and complete assignments/projects.

### **3. Tóm tắt nội dung học phần (Description)**

On completion of this course, learners will be able to understand the software process and apply it to develop a software is some basic knowledge and skills for every software engineer.

Learners will be able to follow a software project as well as to develop a software in a right way based on the software process, from requirement to analysis, design, implementation, and testing.

### **4. Nội dung chi tiết học phần (Outlines)**

#### **Chapter 1 Scope of software engineering**

- 1.1. Basic concepts
  - 1.1.1. Software
  - 1.1.2. Software engineering
  - 1.1.3. Software development
- 1.2. Scope of software engineering
  - 1.2.1. Historic aspect
  - 1.2.2. Economic aspect
  - 1.2.3. Maintenance aspect
  - 1.2.4. Team aspect
  - 1.2.5. Analysis and design aspect
  - 1.2.6. Testing aspect

#### **Chapter 2 Software process**

- 2.1. Technical view
  - 2.1.1. Requirement workflow
  - 2.1.2. Analysis workflow
  - 2.1.3. Design workflow
  - 2.1.4. Implementation workflow
  - 2.1.5. Testing workflow
- 2.2. Unified process
  - 2.2.1. Inception
  - 2.2.2. Elaboration
  - 2.2.3. Construction
  - 2.2.4. Transition
- 2.3. CMM
  - 2.3.1. CMM level 1
  - 2.3.2. CMM level 2
  - 2.3.3. CMM level 3
  - 2.3.4. CMM level 4
  - 2.3.5. CMM level 5

#### **Chapter 3 Software life-cycle models**

- 3.1. Software life-cycle model in theory
- 3.2. Software life-cycle models
  - 3.2.1. Code and fixed
  - 3.2.2. Iteration and increment
  - 3.2.3. Waterfall
  - 3.2.4. Rapid prototype
  - 3.2.5. Agile processing
  - 3.2.6. Spiral

#### **Chapter 4 Testing, planning and estimating**

- 4.1. Testing
  - 4.1.1. SQA
  - 4.1.2. Non-execution testing
  - 4.1.3. Execution-based testing
- 4.2. Planning and estimating

- 4.2.1. LOC
- 4.2.2. FFP
- 4.2.3. Function point
- 4.2.4. CoCoMo

### **Chapter 5 Requirement**

- 5.1. Concept exploration
- 5.2. Business model
  - 5.3.1. Description in natural language
  - 5.3.2. Build general use case for the system
  - 5.3.3. Build detail use case for module
- 5.3. Case study: student project in requirement phase

### **Chapter 6 Analysis**

- 6.1. MVC model
  - 6.1.1. Entity class
  - 6.1.2. Control class
  - 6.1.3. View class
- 6.2. Writing scenarios
  - 6.2.1. Standard scenario
  - 6.2.2. Exception scenario
- 6.3. Extracting classes
  - 6.3.1. Extracting entity classes
  - 6.3.2. Control and view classes
  - 6.3.3. Class diagram
  - 6.3.4. Sequence/collaboration diagram
- 6.4. Case study: student project in analysis phase

### **Chapter 7 Design**

- 7.1. Database design
- 7.2. Architecture design
  - 7.2.1. MVC design
  - 7.2.2. Using CRC card
  - 7.2.3. Class candidate
- 7.3. Detail design
  - 7.3.1. Attribute and method design
  - 7.3.2. Algorithm design (SFM)
- 7.4. Case study: student project in design phase

### **Chapter 8 Implementation and testing**

- 8.1. Code convention
  - 8.1.1. Comment
  - 8.1.2. Naming of variable, method
  - 8.1.3. Code block
  - 8.1.5. Modulo
- 8.2. Unit test
  - 8.2.1. Boundary input variable testing
  - 8.2.3. Unit test tutorial
- 8.3. Functional test
  - 8.3.1. Test case component
  - 8.3.2. Functional test plan
  - 8.3.3. Test case writing
- 8.4. Case study: student project in implementation and testing

## **5. Học liệu (Textbooks)**

### **5.1. Học liệu bắt buộc (Required Textbooks)**

- [1]. Stephen R. Schach. Object-Oriented and Classical Software Engineering. 8<sup>th</sup> edition,

McGraw Hill, 2010.

## 5.2. Học liệu tham khảo (Optional Textbooks)

- [2]. Brahma Dathan, Sarnath Ramnath. Object-Oriented Analysis, Design and Implementation: An Integrated Approach. 2<sup>nd</sup> edition. Springer, 2015.
- [3]. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3<sup>rd</sup> edition, Addison-Wesley Professional, 2018.

## 6. Phương pháp, hình thức kiểm tra – đánh giá kết quả học tập học phần (Grading Policy)

<b>Grading method</b>	<b>Percentage</b>	<b>Group/Individual</b>
- Attendance	10%	Individual
- Exercises	20%	Individual
- Mid-term projects/exams	20%	Group
- Final examination	50%	Individual

**Trưởng Bộ môn  
(Head of Department)**

**Nguyễn Mạnh Hùng**

**Giảng viên biên soạn  
(Lecturer)**

**Nguyễn Mạnh Hùng**