

# Software Engineering in Practice

## Course introduction

Diomidis Spinellis  
Department of Management Science and Technology  
Athens University of Economics and Business

dds@aueb.gr  
<http://www.dmst.aueb.gr/dds>  
@CoolSWEng

2025-02-19

### **What is software engineering?**

#### **Software engineering**

“The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.”

— ISO/IEC/IEEE Systems and Software Engineering Vocabulary (SEVOCAB)

#### **Course objectives**

- Cover of the software engineering body of knowledge
- Experience key professional tools and practices
- Software configuration management
- Building automation
- Issue tracking
- Tool implementation
- Contribution to a real software system
- Practice in
  - software maintenance tasks
  - group assignment
  - collaboration at global level
  - use of popular tools

#### **Course structure**

- Software requirements
- Software design
- Software implementation
- Software testing
- Software maintenance
- Software configuration management

- Software engineering management
- Software engineering process
- Software engineering models and methods
- Software quality
- Software engineering professional practice
- Software engineering economics

## **Course organisation and teaching method**

- Independent study
- Participation
- Questions
- Assignment
- Assignment teams
- Lab and tools
  - Wednesday, 13:00-15:00

## **The notes**

- The course's presentations
- Will not be distributed
- Accessible at <http://www.dmst.aueb.gr/dds/sweng-en>
- (Greek version at <http://www.dmst.aueb.gr/dds/sweng-el>)
- Summary of the course's topics
- Updated continuously
- Printable as one page

## **Lecture schedule**

- Stand-up meeting for each team with respect to the assignment (3')
- Stand-up presentation from each team's representatives (5'–10')
  - All teams or a random subset of them?
- Discussion on topics you studied prior to the lecture
- Short presentation of next lecture's topics and assignment.

## **Grading**

- During the semester
- No exams
- Grade allocation
- Group assignments (30%)
- Use of tools at the lab (20%)
- Class participation (20%)
- Implementation assignment (40%)

(Total 110%)

### **Implementation assignment**

- Contribution to an open source software (OSS) project
- Grade distribution (110%)
- Comprehension and documentation of OSS project (10%)
- Contribution breadth (15%)
- Quality of implementation (15%)
- Integration (10%)
- Testing (10%)
- Collaboration with the OSS project's development team
- Presentations (10%)
- Quality of deliverables (10%)
- GitHub use (wiki, issues, commits, branching) (10%)
- Code reviews (received and submitted) (10%)

### **Choosing an open source project**

- In a language you know or willing to learn
- Runs and builds on a platform you use / can put together
- At least modestly popular (>5-10 stars)
- Developed by a team (>2 contributors)
- Evidence of accepted PRs
- Documented
- Active
- Processes (CI, reviews, issues)
- Avoid algorithm implementations, collections, Greek

### **Excluded and open projects**

- Excluded
  - Jarvis
  - Zulip
  - Sherlock-project
- Looking for contributors
  - sunshineGR
  - python-docs-gr (If you find a code contribution.)
  - Awesome First PR Opportunities

### **Class participation**

- Register the code in the piece of paper through the form at pixel-bonus.com
- You can view participation results on this page

## Course preparation

- Download the SWEBOK from the course's Moodle materials
- The taught chapters of SWEBOK are also available in Greek
- Create a GitHub account
- Form assignment teams of 5-8 students
- Form implementation teams of 1-2 students
- Register in Moodle for this course
- Declare your participation to the course through this form
- Don't forget to declare additional duties that interest you

## Advice for the weekly presentations

- Study the theory; many terms have non-obvious meanings.
- Read beyond SWEBOK, which is a dry, often unengaging, concise guide.
- Do not verbatim read notes or slides when presenting.
- Do not present material that can be easily produced by generative AI. Instead, produce original research through analysis, synthesis, critique, interpretation, comparison, classification, evaluation, triangulation, visualization, experimentation.

## Indicative time schedule

- Milestone 1: Participation and teams (23/2)
- Milestone 2: Selection of OSS project (16/3)
- Milestone 3: Presentation of OSS project (17/3)
- Milestone 4 (50%): Presentation of contribution design (7/4)
- Milestone 5 (final): Presentation of implementation

## Next lecture preparation

- Study SWEBOK v 3.0 chapter 1
- Obtain a GitHub account
- Read the guidelines How to Contribute to Open Source
- Visit the repository How to choose (and contribute to) your first open source project
- Assignment:
- Identify requirements in an important/popular OSS project.
- Specific objectives:
  - Identify stakeholders
  - Identify functional and non-functional requirements
  - Study the impact of change to the requirements
- Video: <http://www.youtube.com/watch?v=VDjoiRkt06M>
- Prepare your participation badge using the provided MS Word DOCX, Open/Libre Office ODT, or Inkscape PDF template.

### **Presentation on OSS**

- Working with open-source software
  - Selecting, using, adapting, and contributing back

### **Distribution License**

Unless otherwise expressly stated, all original material on this page created by Diomidis Spinellis, Marios Fragkoulis, and Antonis Gkortzis is licensed under the Creative Commons Attribution-Share Alike Greece.

