

# II.2405 – Software Engineering

## GENERAL INFORMATION

---

Title: Software Engineering

Module ID: II.2405

Module leader: Zakia KAZI-AOUL

ECTS: 4

Average workload per student: 100 hours, including 42 hours face to face

Teamwork: yes

Keywords: Needs extraction, Development cycle, Testing, Agile approaches, Scrum, Kanban, XP, TDD, Design pattern, UML, Clean code

## PRESENTATION

---

*Computer systems, such as social networks, commercial sites or video games, are an important component of our daily lives. The development of its systems requires the use of methods, approaches, and tooling adapted to the different phases of their life cycle. This results in the application of a systematic, disciplined and quantified approach to the specification, design, development, testing and deployment and maintenance of these systems.*

*In this module, you will become familiar with both traditional methods and agile approaches that enable cost-effective, high-quality IT products that best meet customer requirements. You will also discover the "best practices" in software development.*

## EDUCATIONAL OBJECTIVES

---

- Know how to use the techniques of extracting needs
- Draw up specifications based on customer needs
- Produce a specification file with an agile approach
- Driving a project using agile Scrum and Lean approaches
- Knowing how to apply best practices in software development

## Prerequisite

First experience in team project management, basic notions in Java (Python failing that) and OOP.

## Content/Program

### Concepts

- Methods and processes (cycles, conventional methods, ORs)
- Software quality
- Needs extractions
- Agile approaches (Kanban, XP, continuous improvement, Scrum)
- Design patterns
- Test
- Clean code (SOLID principles, etc.)

### Tools used by the teacher/counsellor

- Practical animation material (Post-it, Lego®, dashboard, etc.)
- Java Development Environment

### Tools used by the learner

- Planning and project management tools (e.g. Jira, Scrum Board)
- Some design patterns (Factory, Strategy, etc.)
- Artifacts from agile approaches (e.g., elevator pitch, impact mapping, story map, user story, etc.)
- JUnit for Test-Driven Development (TDD)

## PEDAGOGICAL METHODS

---

### Learning methods

The module takes place by alternating theory (about 1/3) and practice (about 2/3). Classes can be taught in whole classes or in small groups.

This module includes team project sessions (4 to 5 students per team) as well as practical work sessions. Theoretical concepts are often illustrated by real cases, or can be put into practice immediately through role-playing or exercises.

### Evaluation methods

The evaluation of this module is based on a team project of 4 to 5 students as well as on knowledge tests (individual evaluation).

- Collective evaluation (from 40% to 50%): Draft specification by team
- Individual evaluation (from 60% to 50%): 2 to 3 knowledge tests spread over the semester

### Language of work

The course is taught in English.

Project deliverables can be submitted in English or French. The various knowledge tests are proposed in French and English.

## BIBLIOGRAPHY, WEBOGRAPHY, OTHER SOURCES

---

- **Open class room :**
  - <https://openclassrooms.com/courses/initiez-vous-a-la-gestion-de-projet-agile>
  - <https://openclassrooms.com/courses/gerez-un-projet-digital-avec-une-methodologie-en-cascade>
- **Lean Startup:**
  - <http://theleanstartup.com>
  - <http://theleanstartup.com/book>
- **Story mapping:**
  - <https://www.dunod.com/sciences-techniques/story-mapping-visualisez-vos-user-stories-pour-developper-bon-produit>
  - <https://www.featuremap.co/fr>
- **Kanban:**
  - <https://www.dunod.com/sciences-techniques/kanban-pour-it-une-nouvelle-methode-pour-ameliorer-processus-developpement>
- **Scrum:**
  - <https://www.dunod.com/sciences-techniques/scrum-guide-pratique-methode-agile-plus-populaire>
  - <https://openclassrooms.com/courses/gerez-votre-projet-avec-une-equipe-scrum>
- **User-stories:**
  - <https://www.mountaingoatsoftware.com/books/user-stories-applied>
- **Agile Conferencing**
  - Agile France: [conf.agile-france.org/](http://conf.agile-france.org/)

- Agile Tour: [www.agiletour.org](http://www.agiletour.org)
  - Lean Kanban France: <http://leankanban.fr>
- **Clean code**
  - Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin
- Head First Design Patterns, Freeman, Eric T, Elisabeth Robson, Bert Bates, Kathy Sierra, O'Reilly Media, 2004
- <http://www.sei.cmu.edu/publications/documents/02.reports/02tr011.html>
- [www.swebok.org](http://www.swebok.org)
- Software Engineering, Fourth Edition, Shari Lawrence Pfleeger, Joanne M. Atlee, International Edition, 2010
- Software Engineering: A Practitioner's Approach, Roger S. Pressman. McGraw Hill Higher Education, April 2009
- Software Requirements, Second Edition (Pro-Best Practices), Karl E. Wiegers, Microsoft Press, March 2003