

## Software Development & Operations (E761041)

**Course size** *(nominal values; actual values may depend on programme)*

**Credits 3.0**

**Study time 90 h**

**Course offerings and teaching methods in academic year 2023-2024**

A (semester 1)

Dutch

Gent

lecture

seminar

group work

**Lecturers in academic year 2023-2024**

Volckaert, Bruno

TW05

lecturer-in-charge

**Offered in the following programmes in 2023-2024**

**crdts**

**offering**

[Bachelor of Science in Engineering Technology\(main subject Information Engineering Technology\)](#)

3

A

[Linking Course Master of Science in Information Engineering Technology](#)

3

A

[Preparatory Course Master of Science in Information Engineering Technology](#)

3

A

**Teaching languages**

Dutch

**Keywords**

Computer Science, ICT, Computer technology, DevOps, Continuous Integration, Continuous Deployment, Agile

**Position of the course**

In this course students get to know modern methodologies and supporting tools for developing software-systems in team. This software must adhere to functional and non-functional demands before being put in production. Because software is inherently becoming more complex, there's a need for technologies that keep development and management of software in check.

**Contents**

- Requirement Engineering
  - Functional requirements vs. quality attributes (non-functional)
- Software Development Models
  - Plan-driven software development
  - Agile software development (SCRUM, Kanban, TDD, BDD)
  - DevOps
- Software testing
  - Functional testing (Unit tests, Integration Tests, End-to-End tests, ...)
  - Performance testing (e.g. Gatling)
- Software Version Management
  - Evolution from CVS to Git
  - Basic Git (commits, push/pull, branches, tags, pull requests, forks, code review ...)
- Continuous build/integration/delivery (CI/CD)
  - Build tools: maven, ...
  - Automating tests
  - Containers (Docker) as unit for deployment
  - Provisioning tools (Helm, ansible/chef/puppet, ...)
  - CI/CD Pipelines (Jenkins, Gitlab, ...)
- Software Operations
  - Container Orchestration (Kubernetes)
  - Monitoring (Prometheus, Grafana, ...)

- Software Documentation
  - Generated documentation (Javadoc, OpenAPI & automation)
  - Generating manuals (Markdown, AsciiDoc, MkDocs)

#### **Initial competences**

- Being able to program at an advanced level in an object-oriented language like Java or C#
- Basic knowledge about web technologies and data formats (HTTP, JSON, YAML, ...)
- Basic knowledge Linux

#### **Final competences**

- 1 Student can extract requirements for a software problem.
- 2 Student can choose the right development model for a new software project.
- 3 Student can work in team on a software project by means of e.g. Git.
- 4 Student can deploy a CI/CD pipeline for a software project, capable of automatically testing and deploying software artefacts in production.
- 5 Students has a notion of how software in production can be managed and followed up on.

#### **Conditions for credit contract**

Access to this course unit via a credit contract is determined after successful competences assessment

#### **Conditions for exam contract**

This course unit cannot be taken via an exam contract

#### **Teaching methods**

Group work, Seminar, Lecture

#### **Extra information on the teaching methods**

Lectures, exercises on laptop, group work on laptop

#### **Learning materials and price**

Slides on Ufora

#### **References**

#### **Course content-related study coaching**

Interactive support via Ufora; assistant-guided labs; contact with professor and assistants through mailing list and personally by means of an appointment.

#### **Assessment moments**

end-of-term and continuous assessment

#### **Examination methods in case of periodic assessment during the first examination period**

Written assessment with open-ended questions

#### **Examination methods in case of periodic assessment during the second examination period**

Written assessment with open-ended questions

#### **Examination methods in case of permanent assessment**

Assignment

#### **Possibilities of retake in case of permanent assessment**

examination during the second examination period is possible

#### **Extra information on the examination methods**

First term:

- PE1: written exam with open questions
- NPE1: evaluation end result of the project based on project report

Second term:

- PE2: written exam with open questions
- NPE2: evaluation end result of the project based on project report

#### **Calculation of the examination mark**

- 50% of the final grade is determined by the answers to the written exam
  - 50% of the final grade is determined by evaluation of the project work
- To pass, a student needs to receive at least 9/20 for the PE and NPE. If this not the case and the calculated result is 10 or more, the final grade will be changed and the student receives 9/20.

