Software Testing

2025 - 2026

Is part of the next programmes:

- M0012004 Master of Computer Science: Software Engineering
- M0012005 Master of Computer Science: Data Science and Artificial Intelligence
- M0012006 Master of Computer Science: Computer Networks
- M0048004 Master of Computer Science: Software Engineering
- M0048005 Master of Computer Science: Data Science and Artificial Intelligence
- M0048006 Master of Computer Science: Computer Networks
- M0090004 Master of Teaching in Science and Technology: Computer Science
- M0119000 Master of Digital Business Engineering
- M0119000 Master of Digital Business Engineering
- U0001008 Courses open to exchange students in Sciences

Course Code:	2001WETSWT
Study Domain:	Computer Science

Semester:	2E SEM
Contact Hours:	60
Credits:	6
Study Load (hours):	168
Contract Restrictions:	No contract restriction
Language of Instructions:	ENG
Lecturer(s):	T Serge Demeyer
Examperiod:	exam in the 2nd semester

1. Prerequisites *

speaking and writing of:

- English

extra commentary:

(This course is taught in English)

specific prerequisites for this course

Before starting this course a student must meet the following prerequisites.

- **You have profound experience with programming in an object-oriented language (e.g. C++, Java). Advanced programming constructs (exceptions, threads) contain no secrets for you.**
- **You have hands-on experience with the testing of software (e.g. you wrote unit tests using XUnit).**
- **You are capable of reading and interpreting a design written down in the most commonly used design notations (UML, statecharts, ...)**

- You can demonstrate deep knowledge concerning the formal foundations of computer science (logics, algorithms, complexity theory, finite automata)
- You can specify the pre- and post-conditions and the invariants for a given software module (cfr. *design by contract*).
- You are well aware of the various phases in a software project (requirements, analysis, design, implementation, testing, maintenance).
- You have knowledge about techniques for managing quality during a project (cost estimation, code reviews, metrics)
  The easiest way to satisfy these prerequisites is to have a passing grade for the course Software Engineering (BA 3 Computer Science).

2. Learning outcomes *

- apply white-box and black-box test techniques to build a test-suite;
- assess and improve the coverage of a test suite;
- distinguish between various test automation strategies;
- select the most appropriate test techniques for a given test strategy.

3. Course contents *

De student will acquire experience with torough testing and verification of a software system, to guarantee with a certaing degree of confidence that a given software system meets its specification.
The course has a practical ring to it with a minimal theoretical content (taught as testing best practices; and involving self-training), several lab-sessions (trying out several test techniques and strategies on an existing software system) and industrial guest speakers illustrating how testing is done in real projects.

4. International dimension *

- This course stimulates international and intercultural competences.
- Students use course materials in a foreign language.

- The lecturer invites international guest lecturers.
- Students give presentations in a foreign language.

5. Teaching method and planned learning activities

5.1 Used teaching methods *

Class contact teaching

- Lectures
- Practice sessions
- Laboratory sessions

Personal work

5.2 Planned learning activities and teaching methods

- Theory
- The course starts with three introductory lectures by prof. Demeyer.
- Afterwards we switch to a regime of *flipped class room*, where mini-lectures or recorded on video and students should read provided background material and participate in an on-line quizz.
- Next, the course features a few invited speakers from a testing department / QA department in the software industry. They will share their experiences on testing practices in an industrial context.
- Exercises
- During the exercise session you will experiment with numerous test techniques and strategies on a small yet representative system, namely a simplified version of the famous pac-man game.

A detailed time-schedule is published via a separate web-site:
http://ansymore.uantwerpen.be/courses/software-testing.

5.3 Facilities for working students *

Classroom activities

- no specific facilities

Others

No facilities for working students

6. Assessment method and criteria *

6.1 Used assessment methods *

Examination

- Oral without written preparation

Continuous assessment

- Assignments

6.2 Assessment criteria *

The final grade will be based on four components; (a) and (b) will be assessed during the semester while (c) and (d) will be assessed during the exam.

- (a) Flipped class room:
- Each student will need to study the flipped class room material and participate in a small on-line quizz.
- You must hand in the answers to the quizz.
- (b) Lab-sessions:
- You must finish the lab sessions and hand in the solutions on time via blackboard.
- To be eligible for the final exam, you need to get at least *50/100* for each assignment.
- (c) Theory chapters:
- You will be questioned on one of the lectures (theory + flipped class room). During this questioning you may consult the books (a copy will be present during the exam), and your own notes (which you should bring yourself). A digital copy of all material will be availble.

- Pay special attention to elements like coverage; exit-entry criteria, strategy.
- (d) Guest lecture:
- You will be questioned on one of invited guest lectures (random selection). In particular, you should be able to summarize the main points of a given lecture, to relate the guest lecture to some of the chapters in the book(s), to relate the guest lecture to some of the test process principles and finally to relate the guest lecture to some of the test automation strategies.

Copied Code - Code created by Generative AI

For project work in the lab sessions, students may copy code from the internet, but must do so explicit attribution of the source where it is copied from. Students may use generative AI tools for the code of their lab assignments, but must do so with explicit acknowledgment, i.e. it must be clearly indicated which code was created by which tools. In an oral exam of the assignment or during a contact moment, students are expected to elaborate on their assignment (and how they used generative AI).

Reports written with support from generative AI

Students may use generative AI tools as a tool for the reports expected for the lab assignments, similar to initial search engines such as Google and for checking grammar and spelling. In an oral exam of the assignment or during a contact moment, students are expected to elaborate on their assignment (and how they used generative AI).

7. Study material

7.1 Required reading *

- Software Testing: A Craftsman's Approach, Fourth Edition 4th Edition by Paul C. Jorgensen
- Practical Test Design: Selection of traditional and automated test design techniques by István Forgács, Attila Kovács

These books are available in the library as digital copies.

7.2 Optional reading

The following study material can be studied voluntarily :

All course notes (incl. the presentations done by students and industrial guest lectures) are distributed via a separate web-site. See http://ansymore.uantwerpen.be/courses/software-testing for the public part --- and blackboard for the confidential information

8. Contact information *

Professor:

- Prof. S. Demeyer

Assistant:

- Dr. Onur Kılınççeker
- Dr. Mutlu Beyazit

9. Tutoring

- The lab sessions is the part of the course which requires the most work. We organized it such that most of the work can be done in the lab and that you only need to spend some time at work finishing the report. Block the necessary time in your agendas every week; there is a penalty (= extra work) for late submissions.
- The industrial guest lectures offer a unique opportunity to gain insight into the testing practices as used within today's software industry. Use that opportunity to your advantage and ask focussed questions. Use the examples also in your own lectures.
- Don't skip the lab sessions; and if you do need to skip one of them make sure that you catch up afterwards.