

## Course Outline (as of 22-Jan-2015)

Numerical course code	<b>MTAT.03.159</b>
Title	<b>Software Testing</b>
Faculty/Department	<b>Faculty of Mathematics and Computer Science, Institute of Computer Science, Chair of Software Systems</b>
Amount of credits (1 ECTS = 26 hours)	<b>3 ECTS</b>
Amount of credits (before 31-Aug-2009)	<b>2 CP</b>
CP*1.5 (before 31-Aug-2009)	<b>3</b>
Duration in semesters	<b>1</b>
Final assessment	<b>Differentiated</b>
Course responsible	<b>Dietmar Pfahl</b>
Course instructors	<b>Dietmar Pfahl, Briti Deb, Svetlana Omelkova, Oliver Dalberg, Madhu Tipirishetty</b>
Course language	<b>English</b>
Study levels	<b>Bachelor's studies</b>
Forms of teaching and learning	<b>Lectures (incl. practical work / labs): 32 hours Independent work (incl. e-learning): 46 hours</b>
Study period	<b>Weeks 24-39 First lecture: 12-February-2015 Last lecture: 07-May-2015 Labs (7 groups) start in week 25</b>

### Prerequisites

Compulsory:	MTAT.03.094 Software Engineering (6 ECTS, 4 CP)
Recommended:	MTAT.03.130 Object-oriented Programming (6 ECTS, 4 CP)

### Curricula containing this course

Computer Science (2476)	bac.	2011/12	2012/13	2013/14	2014/15
Information Technology (2477)	bac.	2011/12			
Software Engineering (100864)	mas.	2011/12	2012/13	2013/14	

### Objectives

The course addresses the essential concepts of software quality control and testing and introduces various testing strategies and types of testing. It will also give an overview of different software defects, software defect management, and organizational aspects of software testing.

### Learning outcomes

On successful completion of this course, students will be able to demonstrate knowledge of:

- The role of testing in the software development process
- Test planning and documentation
- Different types of testing techniques
- Static testing and defect estimation

### Course elements

The course will cover:

1. Black-box testing and white-box testing techniques
2. Combinatorial testing
3. Usability testing

4. Certification testing
5. Exploratory testing
6. Reliability growth models
7. Inspections and reviews
8. Static code analysis
9. Test measures
10. Test Maturity Model (TMM)
11. Test organization
12. Test tools

**Schedule (tentative – adjustments will be announced on the course wiki)**

- Week 24: Introduction to Software Testing
- Week 26: Black-box test techniques
- Week 28: White-box test techniques
- Week 30: Static Testing (Inspection) and Defect Estimation
- Week 32: Lifecycle, Documentation, Organisation, Tools
- Week 34: Metrics and Test Process Improvement (Test Maturity Model)
- Week 36: Industry Guest Lecture & Exam Preparation
- Weeks 38 & 39: Exams

**Assessment**

- Labs and associated assignments/reports (homework) – work in pairs (60% of course grade)
- Written exam – individual (40% of course grade)