MTAT.03.159  Software Testing  
2017/2018 spring full-time 3 ECTS

**Faculty/Department**  
chair of Software Engineering (LTAT05)

**Amount of credits (1 ECTS=26 hours)**  
3 ECTS

**Amount of credits (before 31.08.2009)**  
2 CP

**Duration in semesters**  
1

**Final assessment**  
differentiated (A, B, C, D, E, F, not present)

**Syllabus credits**  
3 ECTS

**Lecturers**  
Dietmar Alfred Paul Kurt Pfahl (responsible), Mario Ezequiel Scott, Faiz Ali Shah, Jaanus Jaggo, Margus Luik

**Languages of instruction**  
English

**Minimum number of attendants**  
10

**Maximum number of attendants**  
98

**Study levels**  
bachelor's studies

**Target group and/or preconditions for participation**  
informaatika bakalaureuse 2. aasta üliõpilased

**Forms of teaching and learning and no of hours**  
lectures: 16  practice learning: 16  independent work (including e-learning): 46

**Web-based learning**  
Partially

**Study period (in weeks)**  
24.-39. õppenädalal

---

### Prerequisites

**Compulsory:**  
MTAT.03.094 Software Engineering (6 ECTS, 4 CP)  
or LTAT.05.003 Software Engineering (6 ECTS)

**Recommended:**  
MTAT.03.130 Object-oriented Programming (6 ECTS, 4 CP)

### Curricula containing this course

- Computer Science (2476)  
  bac.  2015/2016  2016/2017
- Conversion Master in IT (144919)  
  mas.  2016/2017

### 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  
- Automated GUI Testing

### Brief description

This course introduces systematic methods for software testing.

### Groups

<table>
<thead>
<tr>
<th>Mark</th>
<th>Limit of attendants</th>
<th>Lecturers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. rühm</td>
<td>20</td>
<td>Margus Luik</td>
</tr>
<tr>
<td>2. rühm</td>
<td>20</td>
<td>Jaanus Jaggo</td>
</tr>
<tr>
<td>3. rühm</td>
<td>20</td>
<td>Margus Luik</td>
</tr>
<tr>
<td>4. rühm</td>
<td>18</td>
<td>Jaanus Jaggo</td>
</tr>
</tbody>
</table>
Schedule

<table>
<thead>
<tr>
<th>Lecture Schedule (tentative - adjustments will be announced on the course wiki)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 24: Introduction to Software Testing</td>
</tr>
<tr>
<td>Week 26: Black-box test techniques</td>
</tr>
<tr>
<td>Week 28: White-box test techniques</td>
</tr>
<tr>
<td>Week 30: Static Testing (Inspection) and Defect Estimation</td>
</tr>
<tr>
<td>Week 32: Lifecycle, Documentation, Organisation, Tools</td>
</tr>
<tr>
<td>Week 34: Industry Guest Lecture (tentative)</td>
</tr>
<tr>
<td>Week 36: Metrics and Test Process Improvement (Test Maturity Model) &amp; Exam Preparation</td>
</tr>
<tr>
<td>Weeks 37-39: Exams</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Practical Session Schedule (tentative - adjustments will be announced on the course wiki)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 25: Debugging</td>
</tr>
<tr>
<td>Week 27: Black-Box &amp; White-Box Testing</td>
</tr>
<tr>
<td>Week 29: Combinatorial Testing</td>
</tr>
<tr>
<td>Week 31: Mutation Testing</td>
</tr>
<tr>
<td>Week 33/34: Automated Web-Application Testing</td>
</tr>
<tr>
<td>Week 35: Static Code Analysis</td>
</tr>
<tr>
<td>Week 37: Document Inspection and Defect Prediction</td>
</tr>
</tbody>
</table>

Lecture materials and course home page

Web-based learning environment

https://courses.cs.ut.ee

Compulsory study materials

http://www.slideshare.net/sachin.mk/practicial-software-testing
see https://courses.cs.ut.ee

Recommended study materials

Brian Hambling; Peter Morgan; Angelina Samaroo; Geoff Thompson; Peter Williams
see https://courses.cs.ut.ee

List of independent works and their instructions

see https://courses.cs.ut.ee

Assessment methods and criteria

<table>
<thead>
<tr>
<th>Home work</th>
<th>Homeworks associated to this course are listed in the course web site: <a href="https://courses.cs.ut.ee/20XX/SW-Test/spring">https://courses.cs.ut.ee/20XX/SW-Test/spring</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Final written assessment</td>
<td>The final exam will aim at assessing the level of knowledge of the theoretical aspects of the course.</td>
</tr>
</tbody>
</table>

Requirements to be met for final assessment

General rule:
- Active participation in labs and associated assignments/reports (homework) - work individually or in pairs (60-70% of course grade)
- Participation in final exam - individual (30-40% of course grade)

The following marks must be achieved to be admitted to the final exam:
- Student must get at least 33% of the maximum possible marks for all labs

The following rules must be fulfilled to not fail the course:
- Student must be admitted to final exam
- In final exam, at least 25% of the maximum possible marks must be achieved
- In total (labs and final exam together), at least 50% of the maximum possible marks must be achieved

For further details, see course wiki page

To obtain final grade, the following is required
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
- Automated GUI testing

Options for taking tests/exams at later date
- Resit

Other information
- see https://courses.cs.ut.ee