Course Outline (as of 07-Feb-2017)

<table>
<thead>
<tr>
<th>Numerical course code</th>
<th>MTAT.03.159</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Software Testing</td>
</tr>
<tr>
<td>Faculty/Department</td>
<td>Faculty of Mathematics and Computer Science, Institute of Computer Science, Chair of Software Systems</td>
</tr>
<tr>
<td>Amount of credits (1 ECTS = 26 hours)</td>
<td>3 ECTS</td>
</tr>
<tr>
<td>Duration in semesters</td>
<td>1</td>
</tr>
<tr>
<td>Final assessment</td>
<td>Differentiated</td>
</tr>
<tr>
<td>Course responsible</td>
<td>Dietmar Pfahl</td>
</tr>
<tr>
<td>Course instructors</td>
<td>Dietmar Pfahl, Kristiina Rahkema, Faiz Ali Shah, Margus Luik</td>
</tr>
<tr>
<td>Course language</td>
<td>English</td>
</tr>
<tr>
<td>Study levels</td>
<td>Bachelor’s studies</td>
</tr>
<tr>
<td>Forms of teaching and learning</td>
<td>Lectures (incl. practical work / labs): 32 hours Independent work (incl. e-learning): 46 hours</td>
</tr>
<tr>
<td>Study period</td>
<td>Weeks 24-39 First lecture: 09-Feb-2017 Last lecture: 04-May-2017 Labs (5 groups) start in week 25</td>
</tr>
</tbody>
</table>

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

Curricula containing this course
Conversion Master in IT (144919) mas. 2016/2017
Software Engineering (100864) mas. 2013/2014 2014/2015

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

Course elements
The course will cover:
1. Black-box testing and white-box testing techniques
2. Combinatorial testing
3. Usability testing
4. Exploratory testing
5. Defect estimation
6. Inspections and reviews
7. Static code analysis
8. Test measures
9. Test Maturity Model (TMM)
10. Test organization
11. Test tools

Schedule (tentative – adjustments will be announced on the course wiki)
- Week 24: Introduction to Software Testing
- Week 25: Black-box test techniques
- Week 26: White-box test techniques
- Week 27: Static Testing (Inspection) and Defect Estimation
- Week 28: Lifecycle, Documentation, Organisation, Tools
- Week 29: Industry Guest Lecture
- Weeks 30-35: no lectures
- Week 36: Metrics and Test Process Improvement (Test Maturity Model) & Exam Preparation
- Weeks 37-39: Exams

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