Study programme(s): Information Technologies

Level: Bachelor

Course title: Software Testing

Lecturer: Gordana Rakić

Status: elective

ECTS: 6

Requirements: none

Learning objectives

The course aims to present and critically analyze current software testing techniques, particularly the importance of formal methods in testing.

Learning outcomes

Minimum: At the end of the course, it is expected that a successful student is able to critically assess the importance of software testing and assess the need and usefulness of formal methods during testing.

Desirable: At the end of the course, it is expected that a successful student developed an integrated approach to software testing and formal theories.

Syllabus

Theoretical instruction

Theoretical basis for testing, structural testing, functional testing, the basis for combining formal methods and testing, formal methods based on the model, testing using finite state machines, testing using process algebra, testing using algebraic specification, mutation testing, testing using UML dynamic models, temporal logic and model of checking models and their role in testing and the process of managing software testing.

Practical instruction

Analysis of study examples.

Literature

1. C. Kaner, J. Falk, H. Q. Nguyen: Testing Computer Software, Wiley, 1999
2. B. Beizer, Software Testing Techniques, International Thomson Press, 1990
3. P. C. Jorgensen, Software Testing: A Craftsman's Approach, second edition, CRC Press, 2004
4. Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled, Model Checking, MIT Press, 1999.
5. Ilene Burnstein. Practical Software Testing. Springer-Verlag, 2003
6. Paul Ammann and Jeff Offutt, Introduction to Software Testing, Cambridge University Press, 2008.

Weekly teaching load				Other:
Lectures: 2	Exercises: 0	Practical Exercises: 2	Student research:	

Teaching methodology

At lectures, classical methodology is applied, through usage of a beam-projector. At exercises traditional methods of teaching are applied, through using a projector, to analyze study examples, as well as practical exercises on computers to practice the principles introduced and to get familiar with the recommended tools. Students build on their knowledge by researching each of the topics and the knowledge is checked through the creation of papers that are presented during and at the end of the course.

Grading method (maximal number of points 100)

Pre-exam obligations	**points**	**Final exam**	**points**
Partial assignments	40	Theoretical exam	40
Final project	20		