Course Description

Software is becoming increasingly complex and there is a growing need for systematic strategies for testing and maintaining software. A 2002 NIST study found that the annual cost of a software error due to inadequate software testing infrastructure is up to 60 billion dollars (about 0.6% of the US Gross Domestic Product) and improving the testing infrastructure can save at least one-third of the cost incurred by software errors, or over 20 billion dollars.

This course will examine the fundamentals of software testing by looking at the main phases in the software testing process, and the different types of software testing methods that are available. Problems that arise from testing different types of software, such as object-oriented, distributed, Graphical User Interfaces and web software will be discussed, and the relevant program analysis techniques will be studied. Topics include, but are not limited to, a general overview of the software testing process, approaches to automatic test case generation, test oracles, coverage analysis to decide when to stop testing, test prioritization, mutation testing, regression testing, and automated software testing tools. We will use both a textbook and research papers as review material for the course. There will also be projects where you will apply the testing methods you learn in the class. No pre-requisites, though parts of the course are likely to be technical in nature.

Course objectives

At the end of the semester a student completing the course should have:

- a solid knowledge of the fundamentals and the state of the art in software testing
- a keen awareness of the open problems in software testing and maintenance
- hands-on experience with using existing software testing tools
- improved skills in reviewing research papers critically and identifying open problems