Advanced Software Engineering

Course Introduction, Spring 2018

Dr. Sebastian Proksch

University of Zurich, Switzerland
Course Overview

**Time:** Monday, 10:15 - 12:00, BIN-2.A.10

**Language:** English

**ECTS:** 6.0 Points

**Target Audience:**
Students in MSc Informatics

**Requirements:**
MBM, SE, Programming skills

**Website:**
http://seal.ifi.uzh.ch/ » Teaching » ASE
Lecturers

Prof. Dr. Harald Gall

Dr. Sebastian Proksch

(… plus a fair bit of guest lecturers)
Organization
Multiple Parts

• Lecture
• Mandatory Reading List
• Programming Project (Group)
• (Oral) Exam
• In addition to lecture slides we will occasionally provide mandatory reading
• Mostly scientific articles
• Part of your homework is to read those
• Will be covered in the exam!
Programming Project

• Apply course contents in practice
• Group Work
• Topic: Replication of an Existing Software Development Tool  
  (More details follow in a minute)
• Groups of 2-3 students
• Two Graded Presentations  
  (Midterm + Final)
(Oral) Exam

• Time to be announced  
  (depends on number of students)

• Individual, 20 - 30 minutes per student

• At least one question per student is on the reading material
Course Topics
Software Architecture
Release Management & Quality Control
Program Representations & Static Analysis
Language Engineering
Mining Software Repositories
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.02</td>
<td>Kickoff and Introduction of Project</td>
</tr>
<tr>
<td>26.02</td>
<td>Software Architecture</td>
</tr>
<tr>
<td>05.03</td>
<td>Testing</td>
</tr>
<tr>
<td>12.03</td>
<td>Program Representations and Parsing</td>
</tr>
<tr>
<td>19.03</td>
<td>Static Analysis</td>
</tr>
<tr>
<td>26.03</td>
<td>Mining Software Repositories</td>
</tr>
<tr>
<td>02.04</td>
<td>No Lecture (Ferien)</td>
</tr>
<tr>
<td>09.04</td>
<td>Midterm Project Presentations</td>
</tr>
<tr>
<td>16.04</td>
<td>No Lecture (Sechseläuten)</td>
</tr>
<tr>
<td>23.04</td>
<td>Release Engineering</td>
</tr>
<tr>
<td>30.04</td>
<td>Optimization and Performance</td>
</tr>
<tr>
<td>07.05</td>
<td>Aspect-Oriented Programming</td>
</tr>
<tr>
<td>14.05</td>
<td>Cloud Applications</td>
</tr>
<tr>
<td>21.05</td>
<td>No Lecture (Pfingstmontag)</td>
</tr>
<tr>
<td>28.05</td>
<td>Final Project Presentations</td>
</tr>
</tbody>
</table>

Check website for updates!
Project: Software Development Tools
Intelligent Code Completion

```csharp
public class MyDialog : Dialog
{
    public override void Create()
    {
        var text = new Text();
        text..
    }
}
```
Typical Workflow

Source Code → Model → Recommender → Developer

Mining → Uses → Proposal → Query
• Replicate a state of the art software development tool.

• You will be provided with a scientific paper that introduces the tool.

• You will be provided with a dataset and a framework that makes it easy for you to analyze source code repositories.

• Make use of techniques and technologies introduced in the lecture.

• A more detailed project description and a list of papers will be uploaded to the course website in the next days.
Technology

JUnit 5

Git

KaVE

http://www.kave.cc

Mockito

Maven

WEKA

The University of Waikato

Travis CI
Deliverables

• Documentation
• Cookbook that explains the design and your steps (before you start)
• Manual with clear instructions how to deploy, run, test (document results)
• Source Code of Fully Implemented Tool
• Describe your approach and point out peculiarities or specific design decisions.
• Discuss open problems.
• Lay out the required steps to implement the tool
• Demo the current state
Final Presentation

- Describe Your Development Process
- Reflect on Design Presented in Midterm
- Demo your approach.
- Report on Your Efforts.

- Maybe, we manage to conduct a “competition” between the teams to find the best tool.
Grading
Graded Parts

**Total:** 100P

Exam: 60P

**Project Results:** 25P

**Project Presentations:** 15P

**Please note:** Barring exceptional circumstances, projects are graded as teams
Grading Criteria

• **5.5-6.0**: An excellent work*
• **5.0-5.5**: A high quality work
• **4.5-5.0**: A good work with just a couple of small weaknesses
• **4.0-4.5**: An average work with clear weaknesses

(*) I will include and acknowledge excellent project results in an upcoming scientific paper.
Presentation Grading

- Structure / clarity of presentation
- Relation to lectures
  (Pick up lecture topics!)
- Sophistication of demos
- Technical sophistication of solution
Project Requirements

- You have successfully re-implemented the assigned tool.
- You provide a set of learned models for the tool.
- You use Git to coordinate team development.
- You have a test suite of unit tests that show the correctness of your code.
- Your tool can be built with Maven.
- A build server is used for continuous integration.
- The result is deployed to a public Maven repository.
- You provide an example client that uses the tool to answer a set of predefined queries.
Next Steps
Next Steps

• Form a team (*now* would be a good time)
• Review Project Proposal and Available Topics (both .pdf files will be uploaded to the course page in the next days)
• Send me an email with your group members and your preferred topics before the next weekend.
• Single registrations are possible, but will be assigned to teams.
• Project Assignment will be announced in the next lecture.