**COURSE DESCRIPTION - ACADEMIC YEAR 2017/2018**

<table>
<thead>
<tr>
<th><strong>Course title</strong></th>
<th>Software Maintenance and Evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Course code</strong></td>
<td>76017</td>
</tr>
<tr>
<td><strong>Scientific sector</strong></td>
<td>INF/01</td>
</tr>
<tr>
<td><strong>Degree</strong></td>
<td>European Master in Software Engineering (LM-18)</td>
</tr>
<tr>
<td><strong>Semester</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Year</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Credits</strong></td>
<td>8</td>
</tr>
<tr>
<td><strong>Modular</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Total lecturing hours</strong></td>
<td>48</td>
</tr>
<tr>
<td><strong>Total lab hours</strong></td>
<td>24</td>
</tr>
<tr>
<td><strong>Total exercise hours</strong></td>
<td>--</td>
</tr>
<tr>
<td><strong>Attendance</strong></td>
<td>Not compulsory</td>
</tr>
<tr>
<td><strong>Course page</strong></td>
<td><a href="https://ole.unibz.it/">https://ole.unibz.it/</a></td>
</tr>
</tbody>
</table>

**Specific educational objectives**
The course belongs to the type "caratterizzanti - discipline "Advanced Topic in Software Engineering" (EMSE - ATSE).

Software systems can be in use for years, if not decades - extremely long time periods during which they must be continuously updated to respond to changes in customer needs or other factors. The goal of this course is to teach students basic and advanced techniques in order to successfully evolve real-world software projects. The course will cover the following key software maintenance and evolution activities:
- Concept location
- Impact analysis
- Actualization
- Refactoring
- Verification

The concepts seen during the lecture will be practiced during a project on a large, established open-source software.

**Lecturer**
Romain Robbes

**Contact**
Piazza Domenicani 3, Room 1.16, RRobbes@unibz.it, +390471 016025

**Scientific sector of lecturer**
INF/01

**Teaching language**
English

**Office hours**
To be defined and published on the web page of the course.

**Lecturing Assistant (if any)**
--

**Contact LA**
--

**Office hours LA**
--

**List of topics**
- Introduction to software maintenance and evolution
- Supporting maintenance and evolution by mining software repositories
- Using software metrics to assess and monitor the quality of software systems
- Software Refactoring
- Using textual analysis techniques in the context of software maintenance and evolution
- Using software metrics for bug prediction
- Search-based algorithms to support maintenance activities
- Release planning: effort and cost estimation techniques

**Teaching format**
Frontal Lectures, in-class exercises, and group project

### Learning outcomes

**Knowledge and understanding**
- Know the main methods and techniques for designing, creating, and maintaining software products and services.
- Know the main methods for (re)engineering, refactoring and optimizing software products and processes.
- Know the main methods of team, resource management and risks analysis in software development and maintenance.

**Applying knowledge and understanding**
- Be able to design and implement information systems in vertical sectors of applications according to technical, functional and organizational requirements.
- Be able to apply innovative methods for management and improvement of development processes in different application domains such web or mobile.
- Be able to design and execute experimental analyses on information systems or their components.

**Making judgments**
- Be able to plan and re-plan a technical project activity aimed at building an information system and to bring it to completion by meeting the defined deadlines and objectives.
- Be able to identify reasonable work goals and estimate the resources required to achieve the objectives.

**Communication skills**
- Be able to coordinate the work of a project team and to interact positively with members of the group.
- Be able to present in a fixed time the content of a scientific/technical report in front of an audience also composed of non-specialists.

**Learning skills**
- Be able to autonomously extend the knowledge acquired during the study course by reading and understanding scientific and technical documentation in Italian, German and English.
- Be able, in the context of a problem-solving activity, to extend even incomplete knowledge taking into account the objective of the project.

### Assessment
The assessment of the course consists of two parts:
- a project (50%);
- an oral exam (50%).

In case of a positive mark the project will count for all 3 regular exam sessions. The projects have to be delivered at least one week before the final oral exam, otherwise they cannot be assessed, and the exam cannot be registered.
<table>
<thead>
<tr>
<th>Assessment language</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation criteria and criteria for awarding marks</td>
<td>The project (50% of the mark), in which a team of students implements a series of change requests on an existing, large-scale, open-source software project, applying the techniques seen in class. Furthermore, progress is documented with state of the art tools, and periodically presented in class. The project will be assessed based on how students apply the techniques seen in class during the project, how their progress is documented, and their ability to work in a team. The project is a group activity; however, each student will be evaluated separately. The oral exam will be assessed based on the acquired knowledge and the understanding of the material presented during lectures, the clarity of answers, mastery of language (also with respect to teaching language), and the ability to summarize, evaluate, and establish relationships between topics.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplementary readings</td>
<td>• Martin Fowler, Refactoring: Improving the Design of Existing Code (Addison-Wesley Professional). ISBN: 0201485672 Additional resources will be made available on the course website on an as-needed basis.</td>
</tr>
</tbody>
</table>

| Software used | The following list includes the most important tools that we will use in the course: • Eclipse IDE • Git • Github |