COURSE DESCRIPTION – ACADEMIC YEAR 2016/2017

Course title	**Empirical Software Measurements**
Course code	76005
Scientific sector	INF/01
Degree	European Master in Software Engineering (LM-18)
Semester	2
Year	1
Credits	8
Modular	No

Total lecturing hours	48
Total lab hours	24
Total exercise hours	--
Attendance	Not compulsory
Prerequisites	Basics of Object-Oriented programming and statistics.
Course page	https://ole.unibz.it/

Specific educational objectives	The course belongs to the type caratterizzanti – discipline informatiche "Advanced Topic in Software Engineering" (EMSE - ATSE). Students will acquire competence in measurement and empirical software engineering. Students will learn how to design, implement, and reflect on experiments with software data and artefacts to build new theory and models of estimation and prediction. In the context of the lab hours, students will also learn how to mine software repositories in order to extract information useful to plan software maintenance and evolution activities.

Lecturer	Xiaofeng Wang
Contact	Piazza Domenicani 3, Room 3.14, Xiaofeng.Wang@unibz.it
Scientific sector of lecturer	INF/01
Teaching language	English
Office hours	Friday 14:00-16:00 – No appointment required Any other day by appointment
Lecturing Assistant (if any)	*TBD*
Contact LA	*TBD*
Office hours LA	*TBD*
List of topics	• Fundaments of measurement theory • Review of existing software measures (internal and external attributes of product, process and resources: measures of size and structure; function points, measures of quality, cost, effort, and software reliability; object-oriented metrics) • Mining Software Repositories • Formulation and testing of hypotheses: expressing hypotheses; selection of a test approach; design and execution of experiments, surveys, and case studies • Experimental design • Results Evaluation: evaluating results with reference to hypotheses. Evaluation of validity: internal and external validity; threats to validity; evaluating the validity of published research results. • Prediction models

Teaching format	Lectures and labs with theoretical and practical exercises. Group projects.

Learning outcomes	Knowledge and understanding: • Have solid knowledge in both foundations and applications of different areas of Computer Science • Be able to analyse and solve complex problems in the area of Software Engineering with special emphasis on the use of the methods and techniques of empirical assessment Applying knowledge and understanding: • Be able to apply empirical analysis fundamentals of ICT data (e.g. Data Mining) for the construction of mathematical models for the evaluation and prediction of the applications' features and software systems • Ability of reading and understanding specific scientific material, such as proceedings of conferences, articles of scientific journals and technical manuals Making judgments • Be able to identify goals, compatible with available time and resources • Be able to identify reasonable work goals and estimate the resources required to achieve the objectives. Communication skills • Be fluent, in written and oral form, in at least one European language other than English, with reference also to the specific specialized vocabulary • Be able to coordinate a project team and identify those activities needed to reach the goals of the project Learning skills • Be able to formulate and validate theories and define new methods through empirical induction

Assessment	Written exam with verification questions and exercises • Theory Mark: 100% final written exam. Project involving the solution of a problem, the critical discussion of the state of the art, the reporting of the project design and achieved results, and a final presentation. • Practice Mark: 100% research report about the group project at the exam date. **Final Mark**: 50% Theory Mark (written exam) + 50% Practice Mark (project).
Assessment language	English
Evaluation criteria and criteria for awarding marks	Relevant for the Theory assessment: clarity of answers, ability to summarize, deep understanding of experimental designs, methodologies, and data analysis techniques. Relevant for the Practice assessment: ability to work in a team, creativity, identification of interesting research questions to investigate in the group project, effectiveness in the presentation of results and lessons learnt.

Required readings	• Experimentation in Software Engineering. C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B.Regnell, A. Wesslén. Springer, 2012 • Software Metrics – A Rigorous & Practical Approach. N. Fenton, S. Pfleeger.
Supplementary readings	Lecture notes and papers will be handed out during the course
Software used	• R • Latex • An IDE of choice • Git