COURSE DESCRIPTION CARD - SYLLABUS

Course name
Quality management and experimental Software Engineering

Course

Field of study
Computing

Year/Semester
1/2

Area of study (specialization)
Software Engineering

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
English

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture

Laboratory classes

Other (e.g. online)

Tutorials

Projects/seminars
60

Number of credit points

5

Lecturers

Responsible for the course/lecturer:
dr inż. Sylwia Kopczyńska

email: Sylwia.Kopczynska@cs.put.poznan.pl

tel. 61 665 2944

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3, 60-965 Poznań

Responsible for the course/lecturer:
dr inż. Mirosław Ochodek

email: Miroslaw.Ochodek@cs.put.poznan.pl

tel. 61 665 2944

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3, 60-965 Poznań

Prerequisites

Student starting this course shall have a basic knowledge concerning Software Engineering, mathematics and statistics. They should have necessary skills to formulate and test simple statistical hypotheses, skills necessary to prepare a short scientific report, and skills necessary to acquire information from given sources of information.

Course objective

1) Provide students the knowledge regarding experimental Software Engineering, especially related to empirical research methods and their theoretical foundations,

2) Provide knowledge regarding quality management, especially related to quality management systems, assessment of processes maturity and their continuous improvement,

3) Develop students' skills related to evaluating methods, tools, and phenomena in Software Engineering using empirical methods,

4) Develop students' skills related to evaluating and improving software development processes within an organization.

Course-related learning outcomes

Knowledge

1. Has advanced and deep knowledge concerning applying experimental methods in Software Engineering.

2. Has advanced and deep knowledge concerning software measurement (GQM+Strategies, measurement scales, types of measures).

3. Has advanced and detailed knowledge regarding reviews and inspections.

4. Has general knowledge regarding quality assurance at the organization level and maturity of software development processes (e.g. ISO 9001, TQM, ITIL).

Skills

1. Is able to acquire knowledge from the literature, also by performing a systematic literature review.

2. Is able to select an appropriate empirical research method to the given research problem / research question (e.g., controlled experiment, case study, survey, meta-analysis).

3. Is able to plan and conduct empirical sctudies and interpret and discuss their results.

4. Is able to conduct code reviews.

Social competences

1. Understands the importance and need of conducting empirical research to develop and evaluate methods and tools supporting software development.

2. Understands the necessity of being honest while describing the results of scientific research.

3. Is aware of the importance of ethical fairness while conducting empirical research (e.g., how to handle sensitive data).

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Eveluation is performed based on how the research project is conducted, based on the research report, the presentations given during the classes, and peer-review of the research of other students (0-100%).

The final grade is determined using the following scale:

- (90%, 100%] - 5.0

- (80%, 90%] - 4.5

- (70%, 80%] - 4.0

- (60%, 70%] - 3.5

- (50%, 60%] - 3.0

- (0%, 50%] - 2.0

Programme content

Quality Management

- quality management (definition of quality, importance of quality, cost of quality),

- continuous improvement paradigm (Plan-Do-Check-Act, TQM),

- quality management systems and ISO 9000 (definition of quality management system, the structure of ISO 9001, the quality management principles in ISO 9001),

- assessment of processes maturity based on CMMI and SCAMPI,

- good practices of service management with ITIL,

- reviews and inspections (definition of review and inspection, inspection and review processes and their results).

Experimental Software Engineering

- empirical research in software engineering (the role of empirical research in evaluation of tools and methods used in software engineering; relationships between observations, laws, and theories; formulating hypotheses and research questions; quantitative and qualitative methods),

- measurement scales (definition and properties of nominal, ordinal, interval, and ratio scales; types of measurement errors),

- controlled experiments (the goals of controlled experiments; experiment definition; context selection; formulating experiment hypotheses; variables selection: dependent and independent variables; sampling; experiment assumptions; data collection; data validation with the use of statistical methods; data visualization and analysis of probability distributions; testing hypotheses with statistical tests; power analysis; software tools supporting the analysis of the experiment data; interpretation and analysis of the results; classification of threats to validity),

- case studies (goals of cases studies; planning a case study; data and evidence collection; analysis of the data collecting within case study; reporting the results of case study),

- meta-analysis (the goals of meta-analysis; systematic literature reviews; planning and preparation of the review protocol; performing a review; documenting results of review),

- surveys (goals of surveys; types of surveys; preparing surveys; evaluating a survey instrument; the analysis of survey results).

During the project classes students are conducting a research project that has to employ at least one of the following empirical methods: controlled experiment, case study, survey, or meta-analysis. The goal of each project is to find an answer to a given research question (preferably in the area of software quality). The results of the project are described in a report and they presented during the classes.

Teaching methods

Multimedia presentations, moderated discussion, case studies analysis, solving tasks provided by tutors, and project.

Bibliography

Basic

1. C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, and A. Wesslen: Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, 2000.

2. Gordon G. Schulmeyer: Handbook of Software Quality Assurance, ISBN-13: 978-1596931862, Artech House Publishers, 2007.

Additional

1. Ochodek, Miroslaw, et al. Improving the reliability of transaction identification in use cases. Information and Software Technology 53.8 (2011): 885-897.

2. Ochodek, Mirosław, and Sylwia Kopczyńska. Perceived importance of agile requirements engineering practices - A survey. Journal of Systems and Software 143 (2018): 29-43.

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,0
Classes requiring direct contact with the teacher	60	2,5
Student's own work (literature studies, planning an empirical study, preparing the report and presentation slides) [1]	65	2,5

[1] delete or add other activities as appropriate