COURSE DESCRIPTION CARD - SYLLABUS

Course name
Software engineering for embedded and mobile systems

Course

Field of study
Computing

Year/Semester
2/3

Area of study (specialization)
Edge Computing

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes

Other (e.g. online)

Tutorials

Projects/seminars

Number of credit points

1

Lecturers

Responsible for the course/lecturer:
Prof. dr hab. inż. Joanna Józefowska
email: joanna.jozefowska@cs.put.poznan.pl
tel. (0-61) 665-2369
Faculty of Computing and Telecommunications
ul. Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

Prerequisites

Basic knowledge in embedded system architecture, strategy of modelling and analysis of microsystems at the physical and software layer and be familiar with he wired and wireless transmission protocols used in microsystems. Programming skills and knowledge of hardware description languages are also required. The student should also have elementary knowledge of reconfigurable technologies, e.g. FPGA programmable logic circuits and programmable microcontroller circuits. In addition, student should demonstrate an attitude of interest in deepening his/her knowledge in the field of embedded systems, with particular emphasis on the methodology of software development for programmable, hybrid and mobile devices.

Course objective

1. Widening theoretical knowledge in the field of software development methodology for embedded systems, providing information about the main problems of the field.
2. Getting to know the literature on the design, development and use of microsystems.

3. Transfer of knowledge in the field of methods and strategies of software design and modeling for embedded systems used in industrial solutions.
4. Presentation of differences between software development for computer systems and embedded systems with an overview of a particular class of systems.
5. Showing the importance of the chosen methods in the field of software development and testing, characterizing the tools used in the software development process for microsystems.
6. Acquainting with limitations in software development, with security issues and rules of using licenses for embedded systems.
7. Discussion of the specificity of the embedded systems market, the main trends in its development, as well as the characterization of the labor market in the indicated topics on a national and global scale.
8. Awareness of legal, economic and technical limitations in the development of the field.

Course-related learning outcomes

Knowledge

The graduate knows: theoretical basis of designing and developing large IT systems, software development techniques for the embedded system industry, issues of developing interdisciplinary systems, the software lifecycles and the tools supporting its development as well as techniques of modelling complex information systems.

Skills

The graduate is able to: formulate functional and non-functional requirements of IT systems being aware of communication problems between teams of experts and the team of software developers, to integrate the knowledge of the hardware and software layers in order to
implement a system dedicated also for non-IT applications, use the methods of project labor-intensity assessment and manage the risk in the implementation of an IT project, choose tools supporting the development of projects for the embedded system industry, assess the complexity of the tasks provided for in the development of software for the embedded systems industry, including interdisciplinary projects with research potential.

Social competences

The graduate is aware of the need to constantly expand his knowledge, especially in the field of software development tools and techniques and is ready to acquire the knowledge necessary for the implementation of IT projects, compatible with the current standards of software development and communication of its modules.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:
Formative assessment:
- based on answers to questions related to material presented during earlier lectures,
- based on presentation of solutions to problems formulated during preceding lectures.
Total assessment:
- verification of assumed learning objectives related to lectures: evaluation of acquired knowledge on the basis of the problem-solving oriented written test; discussion of correct answers in the test;

Additional elements cover: discussing more general and related aspects of the class topic, showing how to improve the instructions and teaching materials.

Programme content

The lecture program covers the following issues: Various classes of embedded systems and comparing their functionality to computer systems; requirements for embedded systems in terms of effective use of hardware resources and selection of the appropriate technological solution dictated by functional and economic aspects; software design, modeling and analysis methods; embedding the application in an existing system depending on its version, along with an overview of the specifics of working with Unix systems and the eCos system; familiarization with the analysis of BOM (Bill Of Materials); presentation of test scenarios and test automation methods; overview of tools used in the software development process: gcc, qemu, gdb, JTAG and version control tools: GIT, SVN; presentation of the existing emulators of hardware platforms and development strategies for large-system projects; discussing the methods of building the user interface along with developing the mechanism of interaction with the user and the environment (UX - User eXperience) taking into account environmental conditions (climatic, mechanical, working time) in the software design process; specification of the principles of creating documentation at various levels of software development (at the stage of specification creation, architecture development, implementation, reconfiguration, testing, servicing); comparison of commercial and open source solutions; discussion of threats and safety rules; presentation of differences in the area of microprocessor microsystems, reconfigurable and hybrid microsystems; outlining the specificity of the embedded systems market and its development trends, as well as the labor market in the field of embedded systems? both nationally and globally; discussion of legal, economic and technical limitations in the development of the field; summary of the main challenges facing the development of embedded systems and microsystems; discussion of the effective use of programming and scripting languages typical of embedded systems (C / C ++, bash, Python); providing examples of mechanisms for expanding embedded systems, especially in terms of creating drivers for peripheral devices and the entire process of preparing the port of the existing system for new hardware platforms.

Teaching methods

Lecture: multimedia presentation, presentation illustrated with examples presented on black/white board.

Tutorials: solving tasks, practical exercises, discussion, teamwork, multimedia showcase

Bibliography

Basic

1. UML: przewodnik użytkownika, Grady Booch, James Rumbaugh, Ivar Jacobson, WTN 2001.
2. The industrial electronics handbook Wilanowski B, Irwin D., Taylor & Francis, 2011.

Additional

1. Real-Time Systems Design and Analysis: Tools for the Practitioner, P. A. Laplante, S. J. Ovaska, Wiley, 2012.

Breakdown of average student's workload

	Hours	ECTS
Total workload	25	1,0
Classes requiring direct contact with the teacher	15	0,5
Student's own work (literature studies, preparation for tutorials, preparation for tests and exam) [1]	10	0,5

[1] delete or add other activities as appropriate