National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Department of Computer Engineering

Management of IT infrastructure projects
Syllabus

Details of the academic discipline

Level of higher education	***First (bachelor)***
Scope of knowledge	*12 Information technologies*
Specialty	*121 Software Engineering* *123 Computer Engineering*
Educational program	*Computer Systems Software Engineering* *Computer systems and networks*
Discipline status	*Selective*
Form of education	*daytime / extramural*
Grade, semester	*3rd year, spring*
Scope of the discipline	*4 credits/120 hours* *Full-time form: 36 hours of lectures, Lab. works 18 hours, SRS 66 h.* *Extramural form: 36 hours of lectures, lab. works 18 hours, SRS 104 h.*
Semester control/control measures	*Test*
Lessons schedule	*According to the schedule for the spring semester of the current academic year at http://rozklad.kpi.ua*
Language of teaching	*Ukrainian*
Information about the head of the course / teachers	Lecturers: Associate Professor of the Department of CE, Doctor of Technical Sciences, Associate Professor Iryna Anatoliyivna Klymenko, ikliryna@gmail.com Professor of the Department of CE, Doctor of Technical Sciences, Associate Professor Valentyna Vasylivna Tkachenko, tkavalivas@gmail.com Laboratory staff: Artem Volodymyrovych Kaplunov, art.kaplunov@gmail.com Anatoly Ruslanovych Gaidai, tolya.hei@gmail.com https://classroom.google.com/c/NTgwNDU1NTQ3ODgz?cjc=qzdv3d4
Placement of the course	The training course is hosted on the "Sikorsky" distance learning platform in the Google Workspace for Education environment

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

The goal of the discipline "Management of IT infrastructure projects" is to acquire theoretical knowledge and practical skills of IT project management in the scope of development of embedded systems, as well as teamwork skills: acquisition of soft skills by applying hard skills to solve practice problems. The goal of studying the discipline is to learn the modern practices of leading IT enterprises in accordance with world standards.

The study of the discipline is implemented in the form of organization and implementation of a startup project in the field of development of embedded systems. The result of working out the tasks of the discipline is a completed practical project. The content of the discipline considers and involves mentors who are practitioners of leading IT companies.

Gaining knowledge and skills while studying the discipline provides a perfect understanding of modern processes of team development of complex software and hardware systems and complexes at the level of advanced national and world practices, as well as provides the necessary experience and learning for participation in such projects and their implementation in the future.

Studying the discipline strengthens the following general and professional competencies:

- ЗК2. Ability to learn and study modern knowledge
- ФК1. The ability to apply the legislative and regulatory legal framework, as well as state and international requirements, practices and standards in order to carry out professional activities in the field of computer engineering.
- ФК2. Ability to use modern methods and programming languages to develop algorithmic and software.
- ФК5. Ability to use design automation tools and systems to develop components of computer systems and networks, Internet applications, cyber-physical systems, etc.
- ФК9. Ability to systematically administer, use, adapt and operate existing information technologies and systems.
- ФК13. The ability to solve problems in the field of computer and information technologies, to determine the limitations of these technologies.
- ФК14. Ability to design systems and their components considering all aspects of their life cycle and mission, including creation, configuration, operation, maintenance and disposal.
- ФК16. Ability to design, implement and maintain high-performance parallel and distributed computer systems and their components using FPGA modules and automated design systems

In accordance with the above, strengthened general and professional competencies would provide the following learning outcomes:
- gain experience in team implementation of projects using all phases of the life cycle (from gathering requirements to implementation);
- learn how to conduct the pre-project R&D stage of design;
- learn how to develop a technical proposal and technical task;
- learn to conduct requirements engineering (business analysis);
- get practical DevOps experience;
- get practical experience of accompanying; organization of processes (life cycle models) according to classic and Agile models;
- gain experience in implementing projects using different development methodologies (classical and Agile) and in different roles (developer, tester, analyst, DevOps, manager, etc.);
- get practical experience of quality assessment (verification and testing);
- gain practical experience in finding errors in the software and hardware of embedded systems;
- gain experience in using automated means of team development of such projects;
- get practical experience of reporting and handing over the project;
- gain experience in business communication (meetings, presentations, active listening, written communication, interviewing, etc.);
- gain experience working in a team (giving feedback, sharing experience, resolving conflict situations, etc.);
- gain time management experience;
- gain experience in practical solutions to problems;
- gain experience defending their own decisions in a professional discussion and presenting the results of their own developments;
- gain experience in teamwork on complex technical projects in accordance with modern requirements for processes and technologies for their implementation;
- learn how to make an effective CV, present your personal qualities and pass interviews.

2. Pre-requisites and post-requisites of the discipline (a place in the structural and logical scheme of training according to the relevant educational program)

When studying the discipline "Management of infrastructure IT projects", it is advisable to use the knowledge gained during the study of previous disciplines: "Introduction to the Linux operating system", "Computer architecture. Part 1. Control and arithmetic devices", "Computer architecture. Part 2. Processors"; disciplines from the F-catalogue of optional disciplines included in the certificate program: "Programming technologies in C for embedded systems", "Programming technologies in FPGA (FPGA)".

The discipline is basic for courses from the F-catalogue of optional disciplines included in the certificate program: "Technologies of testing (QA) of embedded systems", "Technologies of designing intelligent systems" (selective), and is also useful for studying normative disciplines "Architecture computers Part 3. Microprocessor devices", "Computer architecture. Course work", "System programming", "Computer networks", "Computer systems".

3. Structure of the credit module

SECTION I. ORGANIZATION OF IT PROJECT MANAGEMENT PROCESSES.
Lecture 1. Introduction to IT project management in the scope of embedded systems development
Structure and tasks of the discipline.
Requirements of the IT industry for soft skills.
Basic concepts of IT project management.
Lecture 2. IT project management models.
Models of the life cycle of complex hardware and software complexes.
Agile life cycle models: Agile, Scrum; Kanban.
IT project management toolkit.
Lecture 3. IT project management processes at the R&D stage
Goals, subjects, objects of the R&D stage.
Artifacts of the R&D stage.
Lecture 4. Team organization and teamwork in the process of IT project management
Team and roles on IT projects.
Selection of the team, development and study of CVs.
Collective work at IT enterprises.
Lecture 5. IT project management at the requirements engineering stage
Theoretical foundations of requirements engineering.
Methods of formal requirements analysis and requirements specification techniques.

SECTION II. IMPLEMENTATION OF IT PROJECT MANAGEMENT PROCESSES.
Lecture 6. Creation of effective architectural solutions of software systems
Theoretical foundations of architectural design of complex hardware and software complexes (IoT).
Technologies for creating effective architectural solutions.
Lecture 7. Teamwork in IT projects
The role of unity in the team.
Team effect and its role in IT project management.
Lecture 8. Design and implementation of human-machine interaction processes (UX/UI design)
Principles of human-machine interaction in complex hardware and software complexes (IoT)
Practical aspects and artifacts of UX/UI design
Lecture 9. Collective development of complex hardware and software complexes (IoT)
Collective development of software systems from the standpoint of system analysis.
Assembly programming.
"Clean" code in collective work.
Lecture 10. Making decisions in a group - IT team
Theoretical information of group decision-making: theory of managerial decision-making; group decision making.

Practice of making individual and group decisions.

Lecture 11. Assessment and quality assurance (QA) in IT project management processes

Quality, efficiency, verification, testing - processes and artifacts.

Quality of software and hardware systems.

Lecture 12. Internal processes of team management on IT projects. Practices of leading IT companies

Leadership

Conflictology: compromise and consensus.

Time management: team and personalities.

Lecture 13. Preparation and conducting of effective collective events

Meetings and negotiations.

Effective presentation.

Lecture 14. Implementation and maintenance of complex hardware and software complexes (IoT)

Practice of leading IT companies.

Implementation and support processes.

Practice of implementation and support.

Lecture 15. Development directions of IT project management technologies in the scope of embedded systems in the conditions of the 4th industrial revolution (Industry 4.0/IoT)

Modern IT projects and their management.

IT projects perspectives and their management.

4. Study resources and materials

Base literature

1. Architecture of computers 3. Microprocessor tools. Part 2: Programming for STM32 microcontrollers. Study guide for degree candidates: Laboratory workshop [Electronic resource] : study guide for the student's educational program "Computer Systems and Networks" in specialty 123 "Computer Engineering" / I. A. Klymenko, V. V. A. Taraniuk, V. Tkachenko, A. V. Kaplunov; Igor Sikorsky Kyiv Polytechnic Institute - Electronic text data (). - Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, 2022. - 125 р.Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 1 від 02.09.2022 р.).
2. Darnall Russell W., Preston John M. Project Management from Simple to Complex. 2012. – . https://saylordotorg.github.io/text_project-management-from-simple-to-complex-v1.1/index.html
3. Adrienne Watt Project Management. 2014. – https://www.opentextbooks.org.hk/system/files/export/15/15694/pdf/Project_Management_15694.pdf
4. GitLab Documentation. 2022. – https://docs.gitlab.com/

Additional literature

1. Lavrishcheva K.M. Software Engineering. - K.: 2008. - 319 p.
2. Managing Successful Projects with PRINCE2 (2009 Edition) - Office of Government Commerce, 2009
4. 8. Eric S. Norman. Work Breakdown Structure. ISBN 9780470177129; 2008 - 304 p.

Information resources

The remote course is on the "Sikorsky" distance learning platform in the Google Workspace for Education environment.

Study content

5. Methods of mastering an educational discipline (educational component)

Distribution of study time by types of classes and tasks in the discipline according to the working study plan. 120 hours and 4 credits are allocated to the credit module - 18 lectures (36 hours), 9 laboratory works (18 hours)

To achieve the goal of the educational discipline, the lecture material should focus on managing IT projects in the field of developing embedded systems, as well as teamwork skills: acquiring soft skills by applying your hard skills to solve practice problems.

6. Laboratory works

The goal of laboratory works is to acquire practical IT project management skills in the scope of embedded systems development, as well as teamwork skills within such projects. The purpose of laboratory works is to fully understand the modern processes of team development of a complex system in practice, to obtain the necessary experience and skills to participate in such projects as a developer, as well as to obtain basic knowledge for the organization and implementation of projects in IT companies in various roles, like DevOps.

Laboratory works are organized as the execution and implementation of a startup project in the scope of development of embedded systems. The result of working out the tasks of the discipline is a completed practical project.

Topics of laboratory works

Laboratory work 1. Setting up a working project. Reviewing requirements and adding new requirements
Laboratory work 2. Creation of a sprint and its evaluation
Laboratory work 3. CI/CD Jenkins
Laboratory work 4. Writing code in a team
Laboratory work 5. Code review in the team
Laboratory work 6. Writing unit tests
Laboratory work 7. Launching the automatic build of the code in the team
Laboratory work 8. Conducting a retrospective

7. Self-study independent work

Types of independent work (66 hours):
− preparation for classroom classes (0.5 hours x 18 lectures = 9 hours);
− preparation and processing of calculations based on primary data obtained in laboratory classes, performing laboratory work, solving problems, sending results to GitLab (recommended 1.5 hours x 8 laboratory works = 12 hours);
− execution of modular control work (2 MCW x 4 hours = 8 hours);
− preparation for express tests (4 hours);
− preparation for the final test (4 hours);
- development of topics for independent work, downloading and gathering the software for laboratory work (29 hours).

Topics for self-study (daytime study)

Topic 1. Requirements of the IT industry for flexible skills (soft skills).
Topic 2. IT project management toolkit.
Topic 3. Selection of the team, development and study of CVs.
Topic 4. Collective work at IT enterprises.
Topic 5. Technologies for creating effective architectural solutions.
Topic 6. Conflictology: compromise and consensus.
Topic 7. Meetings and negotiations.
Topic 8. Modern IT projects and their management.
Topic 9. IT projects perspectives and their management.

5. Methods of teaching the discipline in the correspondence form of education

Content of lectures and independent work

Lecture 1.

SECTION I. ORGANIZATION OF IT PROJECT MANAGEMENT PROCESSES.

Introduction to IT project management in the field of embedded systems development

Structure and tasks of the discipline.

Basic concepts of IT project management.

IT project management models.

Models of the life cycle of complex hardware and software complexes.

Flexible - Agile life cycle models: Scrum; Kanban.

Topics for self-study

Requirements of the IT industry for flexible skills (soft skills).

IT project management toolkit.

Lecture 2.

IT project management processes at the R&D stage

Goals, subjects, objects of the R&D stage.

Artifacts of the R&D stage.

Team organization and teamwork in the IT project management process.

Team and roles on IT projects.

IT project management at the requirement engineering stage

Theoretical foundations of requirements engineering.

Topics for self-study

Selection of the team, development and study of CVs.

Collective work at IT enterprises.

Methods of formal requirements analysis and requirements specification techniques.

Lecture 3.

SECTION II. IMPLEMENTATION OF IT PROJECT MANAGEMENT PROCESSES.

Creation of effective architectural solutions of software systems.

Theoretical foundations of designing the architecture of complex hardware and software complexes (IoT).

Teamwork of the team on IT projects

The role of unity in the team.

Design and implementation of human-machine interaction processes (UX/UI design)

Practical aspects and artifacts of UX/UI design

Topics for self-study

Technologies for creating effective architectural solutions.

Team effect and its role in IT project management.

Principles of human-machine interaction in complex hardware and software complexes (IoT)

Lecture 4

Collective development of complex hardware and software complexes (IoT).
"Clean" code in collective work.
Decision-making in the group - IT team.
Practice of making individual and group decisions.
Assessment and quality assurance (QA) in IT project management processes.
Quality, efficiency, verification, testing - processes and artifacts.
Quality of software and hardware and software systems.
Internal processes of team management on IT projects. Practice of leading IT companies.
Time manager: team and personalities.
Implementation and maintenance of complex hardware and software complexes (IoT). Practice of leading IT companies.
Implementation and support processes.
Practice of implementation and support.

Topics for self-study

Collective development of software systems from the standpoint of system analysis.
Assembly programming.
Theoretical information of group decision-making: theory of managerial decision-making; group decision making.
Leadership.
Conflictology: compromise and consensus.
Preparation and implementation of effective collective events.
Meetings and negotiations.
Effective presentation.
Development directions of IT project management technologies in the field of embedded systems in the conditions of the 4th industrial revolution (Industry 4.0/IoT)
Modern IT projects and their management.
IT project perspectives and their management.

Topics of laboratory works for self-fulfillment

Laboratory work 1. Setting up a working project. Reviewing requirements and adding new requirements
Laboratory work 2. Creation of a sprint and its evaluation.
Lab 3. CI/CD Jenkins
Laboratory work 4. Writing code in a team.
Laboratory work 5. Code review in the team

Topics of laboratory works for classroom fulfillment

Auditory lesson 1:

Defending the laboratory works 1-5. Testing.

Auditory lesson 2:

Laboratory work 6. Writing unit tests

Auditory lesson 3:

Laboratory work 7. Starting the automatic assembly of the code in the team

Auditory lesson 4:

Laboratory work 8. Conducting a retrospective

Types of self-study for PhD students (104 hours):

− preparation for classroom classes (1.5 hours x 4 lectures = 6 hours);

− preparation for express tests (4 hours);

− self-fulfillment of laboratory works 1-5 (3.5 hours x 5 laboratory works = 17.5 hours);

− preparation for the fulfillment of laboratory works (6-8), creating protocols, placing protocols on GitLab (1.5 hours x 3 laboratory works = 4.5 hours)

− preparation for MCW1 (4 hours);

− implementation of MCW2 (4 hours);

− preparation for the test (4 hours);

− self-processing of theoretical material, downloading and assembly of software for the fulfillment of the laboratory work (60 hours).

Policy and control

5. Policy of educational discipline (educational component)

Deadlines are set for the performance of laboratory work and modular control work.

Performance of laboratory work outside of the established deadlines is accompanied by penalty points, which are deducted from the grade for the protocol. Modular control work is not accepted beyond the set time.

Penalty points are issued for: untimely submission of laboratory work. The number of penalty points is no more than 10.

Bonus points are awarded for: active participation in lectures; completing current homework, keeping a summary, preparing a message with a presentation on one of the topics of the SRS discipline, etc. The number of bonus points is no more than 10.

Some lecture topics are accompanied by short express tests (for 15 minutes), which include the material of the studied topic and questions that are asked for independent study. The points obtained for the test are included in the semester rating. Current tests are not retaken.

The performance of each laboratory work is preceded by the completion of an individual task and its preparation in the form of a protocol. A student who came to class without a completed protocol is not allowed to do laboratory work. In the first stage, the student defends the results obtained during the performance of an individual task for laboratory work, in the second stage - defends the theory through an oral survey or test. Most of the laboratory works are accompanied by tests to evaluate the studied theoretical and practical material for the laboratory work. The points obtained for the performance of laboratory work, for the test and for the protocol are included in the assessment for the laboratory work. Testing is carried out in a laboratory session after checking the results of laboratory work. A student who has not completed the individual task before the laboratory work and the test is not admitted.

Performance of laboratory work is mandatory for admission to semester control. The condition of admission to the semester control is the enrollment of all laboratory works and a starting rating of at least 30 points.

A modular test is written during a lecture session without the use of aids (mobile phones, tablets, etc.); the result is forwarded to the corresponding Google Drive directory via a Google form.

The modular control paper is not rewritten in case of a negative grade, a negative grade for the MCR (less than 9 points (<60%)) is equal to 0 points, in this case the modular control work is not counted.

The grade that a student can receive for each laboratory work and for each modular control work is given in table 1 of semester work evaluations, chapter 8 of the syllabus.

Thus, the minimum grade that a student must receive for admission to the semester exam is 60 points, the maximum is 100 points for the completion of all current works for the semester.

Applicants who have fulfilled all admission requirements (completed all laboratory work) and have a rating of less than 60 points, as well as applicants who wish to improve their rating, have the opportunity to pass a semester test in the form of a credit test at the last class on the schedule.

In the case of performance of credit control work, the rating is defined as the sum of points for credit control work and points for individual semester tasks.

The individual work of the student related to the performance of laboratory work is included in the individual semester tasks. The maximum number of points for individual work per semester is 60 points. The maximum mark for the test is 40 points. In this way, the applicant has the opportunity to increase his rating by writing a final test and adding additional points to the number of points received during the semester for individual semester work.

After completion of the credit control work, if the grade for the credit control work is higher than the rating, the applicant receives a grade based on the results of the credit control work. If the grade for the final test is lower than the rating, the applicant's previous rating (with the exception of points for the semester individual task) is canceled and he receives a grade based on the results of the final test. This option forms a responsible attitude of the applicant towards deciding on the completion of the credit control work, forces him to critically assess the level of his training and carefully prepare for the credit.

6. Types of control and rating system for evaluating learning outcomes (RSE)

The student's semester rating from the credit module is calculated based on a 100-point scale. The rating consists of the points that the student receives for completing 8 laboratory works R_L, two modular control works R_{MCW} and expert tests R_{ET}.

The maximum number of points for laboratory work is 60 points, i.e R_L =60.

The generalized criteria for evaluating laboratory work are as follows:

- the timeliness of the preparation of the protocol for the laboratory session, completeness of the theoretical or practical task in the protocol, the protocol is posted on GitLab on time;
- correct functioning of the developed models on software or hardware, demonstration of own repository on GitLab with laboratory work materials and availability of commits;
- a survey on the subject of laboratory work for crediting the practical part of the work, protection of the results obtained in the work, answers to additional theoretical questions of the teacher, completeness of the report/protocol on the work on GitLab.

A detailed approach to the assessment of each laboratory work is given in Table 1.

Table 1. Details of the evaluation of each laboratory work

The name of the class	Form of control	Scores	Admission to the	Total points

			exam by automatic evaluation	
Laboratory work 1.	Entrance test	4	2	4
Laboratory work 2	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab	2		
Laboratory work 3	Completing the task	3	5	8
	Polling on QA	3		
	Protocol on GitLab	2		
Laboratory work 4	Completing the task	3	5	8
	Polling on QA	3		
	Protocol on GitLab	2		
Laboratory work 5	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab	2		
Laboratory work 6	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab / demonstration	2		
Laboratory work 7	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab / demonstration	2		
Laboratory work 8	Completing the task	3	5	8
	Polling	3		
	Protocol on GitLab / demonstration	2		
Number of points for individual work				60
Express tests at lectures	2 x 5	10	5	10
Modular control work	MCW1 (Тест)	15	9	15
	MCW2	15	9	15
Total points		100	60	100

The maximum number of points per MCW R_{MCW} = 2x15 = 30 points.

MCW1 is conducted in the form of automated testing on the Google Workspace for Education platform. The test consists of 60 questions R_{MCW_2} = 0,25x60 = 15 points

Modular control work MCW2 is performed independently according to an individual task. MCW2 assessment criteria at four levels:

- correct and meaningful answer with explanations in the terms of the subject area: 13 - 15 points;
- correct answer, incomplete explanations: 11 - 12 points;
- the answer contains errors: 9 - 10 points;
- the answer contains significant errors, there are no explanations: 4-8 points;
- no answer: 0 points.

The score for MCW2 is reduced by:

- incorrect registration;
- lack of comments in meaningful terms;

– lack of explanations during calculations.

The maximum number of points for express tests is 10 points, tests are conducted during lectures in the form of automated testing on the Google Workspace for Education platform.

The maximum number of points for the credit control work is R_T = 40 points.

The credit control work is conducted in the form of automated testing on the Google Workspace for Education / moodle platform, consisting of selected questions that were during the semester in MCW, express tests, and defenses of laboratory works. The maximum score for the credit control work R_T = 40 points.

Calendar certification of students (for 8 and 14 weeks of semesters) in the discipline is carried out according to the value of the student's current rating at the time of certification. If the value of this rating is at least 50% of the maximum possible at the time of certification, the student is considered certified. Otherwise, the attestation information is marked as "uncertified".

The number of points a student receives per semester is determined by the formula

$$\mathbf{R = R_L + R_{MCW} + R_{ET}.}$$

The maximum number of points per semester does not exceed RS = 100.

Considering the received sum of points, the final grade is determined according to table 3.

If a student writes a test paper, the number of points the student receives per semester is determined by the formula

$$\mathbf{R = R_{IP} + R_T}$$

where, $\mathbf{R_{IP} = R_L}$.

The maximum number of points per semester does not exceed R = 100.

Considering the received sum of points, the final grade is determined by table 3.

Table 2. Determination of the semester grade

Scores	*Rating*
100-95	Perfectly
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

Working program of the academic discipline (syllabus)

Made by, Ph.D., associate professor, professor of the department of CE Klymenko Iryna Anatoliivna, assistant of the department of CE, Taraniuk Victoria Anatoliivna.

Approved by the Department of Computer Engineering (Protocol No.10 dated 25.05.2022 p.)

Agreed by the methodical commission of FICT (protocol No.10 dated 09.06.2022 p.)