Syllabus
Course Program

Software Modeling and Analysis

Specialty 121 – Software Engineering	Institute Institute of Computer Science and Information Technology
Educational program Software Engineering	Department Software Engineering and Management Intelligent Technologies (321)
Level of education Bachelor's level	Course type Special (professional), Mandatory
Semester 8	Language of instruction English, Ukrainian

Lecturers and course developers

Oleksandr Shmatko
oleksandr.shmatko@khpi.edu.ua
PhD., Senior Lecturer, Ass. Professor of the Department of Software Engineering and Intelligence Technologies of NTU "KhPI"

Prepared and published more than 90 scientific and educational works Google Scholar: https://scholar.google.com/citations?user=Wyv6ESUAAAAJ
ORCID: https://orcid.org/0000-0002-2426
Scopus: https://www2.scopus.com/authid/detail.uri?authorId= 6602623478
More about the lecturer on the department's website

General information

Summary

The task of the discipline "Software Modelling and Analysis" is to train students in the methodology of System Analysis and modelling, which allows them to solve the following main tasks at the stage of creating software: providing the necessary software functionality and adaptability to the constantly changing conditions of its functioning; designing data objects implemented in the system; designing interface tools (screen forms, reports) that will ensure the execution of data requests; choosing a specific environment or technology for project implementation.

Course objectives and goals

Students' acquisition of the necessary level of knowledge of the basic processes, phases, and iterations of the software life cycle; obtaining practical skills in applying appropriate mathematical concepts of domain, system, and object-oriented analysis methods and mathematical modelling for software development; obtaining knowledge and skills to use methods and tools for collecting, formulating, and analysing software requirements; ability to select source data for design, guided by formal methods of describing requirements and modelling; as well as practical skills in documenting and presenting software development results.

Format of classes

Lectures, laboratory classes. Current control - laboratory work, intermediate modular control. Final control - exam.

Competencies

K01. Ability to think abstractly, analyze and synthesize.
K02. Ability to apply knowledge in practical situations.
K05. Ability to learn and master modern knowledge.
K06. Ability to search, process and analyze information from various sources.
K14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and processes of functioning.
K15. Ability to develop architectures, modules and components of software systems.
K16. Ability to formulate and ensure software quality requirements in accordance with customer requirements, terms of reference and standards.
K17. Ability to comply with specifications, standards, rules and guidelines in the professional field when implementing life cycle processes.
K19. Knowledge of data information models, ability to create software for storing, extracting and processing data.
K23. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate software development models and approaches.
K26. Ability to think algorithmically and logically.

Learning outcomes

PLO01. Analyze, purposefully search and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology.
PLO03. Know the basic processes, phases and iterations of the software life cycle.
PLO05. To know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modeling for software development.
PLO09. To know and be able to use methods and tools for collecting, formulating and analyzing software requirements.
PLO11. Select input data for design, guided by formal methods of requirements description and modeling.
PLO14. Apply in practice software tools for domain analysis, design, testing, visualization, measurement and documentation of software.
PLO23. Be able to document and present the results of software development..

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 20 hours, laboratory classes - 20 hours, self-study - 80 hours.

Course prerequisites

Programming basics
Fundamentals of software development
Computer architecture and operating systems

Features of the course, teaching and learning methods, and technologies

Teaching and learning methods:
interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback, problem-based learning.
Forms of assessment:
written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS).

Program of the course

Topics of the lectures

Topic 1: Basic Concepts of Software Design Technology

Software lifecycle. Subject and object of the course "Software Modelling and Analysis" The concept of Information System software IP classes. Main features of modern software projects Stages of software development include requirements formation, conceptual design, application specification, model development, integration, and testing of the information system. Methods of software engineering in software design Life cycle processes: main, auxiliary, organisational Content and interrelationship of life cycle processes Life cycle models: cascade, intermediate control model, spiral Stages of the software lifecycle. Regulation of design processes in domestic and international standards.

Topic 2: Organisation of Information Systems Software Development

Canonical design. stages of the canonical software design process. Goals and objectives of the pre-project stage of software creation Models of the organisation's activities ("AS-IS" and "TO-BE"). Scope of work at the stage of technical and operational design Composition of project documentation. Standard design. The concept of a standard project, prerequisites for typing objects. Standard design methods. Evaluating the effectiveness of using standard solutions Standard design solution (SDS). SDS classes and structure Composition and content of operations in standard element IP design. Functional application software packages (FAS) as the basis of SDS. Adaptation of a typical Information system Methods and tools for prototyping an information system.

Topic 3: Basic Concepts of Organisational Business Modelling

The company's mission, goal tree, and strategy for achieving them. Static description of the company: business potential of the company, functionality of the company, areas of responsibility of management. Dynamic description of the company. In the process of streaming models. Models of data structures. Complete business model of the company. Templates for organisational business modelling. Building the organisational and functional structure of the company Stages of developing regulations for the organisational and functional structure of the company. Information technologies for organisational modelling.

Topic 4: Specification of Functional Requirements for Information Systems Software

In the process of streaming models. Process approach to organising an organisation's activities. Relationship of the concept of the process approach to the concept of matrix organisation. The main elements of the process approach are: process boundaries, key roles, goal trees, function trees, and indicator trees. Selection and classification of processes. Basic processes, management processes, and support processes Reference models. Conducting a pre-project survey of the organisation. Questionnaires, interviews, and photos of staff working hours. Results of the pre-project survey

Topic 5: Methodology of Modelling the Subject Area

Methodology for modelling the subject area. Structural model of the subject area. Object structure. Functional structure. Management structure. Organisational structure. Functional-oriented and object-oriented methodologies for describing the subject area. IDEF functional diagram. Functional diagram of data flows Object-oriented methodology. Comparison of existing methods.

Topic 6: Information support of Information System

Information support for IS. Out-of-machine information support. Basic concepts of Information classification. Concept and basic requirements for the information encoding system. Composition and content of classifier design operations. Documentation system. In-machine information support. Design of screen forms of electronic documents. Information base and ways to organise it.

Topic 7: Information Support Modelling

Data modelling. The IDEF1 method Displaying the data model in Model display levels. Creating a logical data model: logical model levels; entities and attributes; relationships; entity types and inheritance hierarchy; keys; data normalisation; domains. Creating a physical model: Physical Model levels; tables; validation rules and default values; indexes; triggers and procedures; designing data warehouses; calculating database size; forward and reverse engineering

Topic 8: Object-Oriented Analysis and Design of Information Systems Software

The main types of UML diagrams used in the design of Information systems Relationships between charts. UML support for the iterative IS design process. IS design stages: modelling of Business Use Cases, development of a business object model, development of a conceptual data model, development of

System Requirements, analysis of requirements and preliminary design of the system, development of database and application models, and design of the physical implementation of the system

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Laboratory work #1: Identification of high-level requirements. Development of the vision document
Laboratory work #2: Identifying user requirements. Search for actors and use cases.
Laboratory work #3: Description of use cases (Use-case, UC)
Laboratory work #4: Development of the Glossary
Laboratory work #5: A detailed description of key UC
Laboratory work #6: Analysis and Specification of Special Requirements
Laboratory work #7: SRS Formation
Laboratory work #8: SRS verification

Self-study

Individual assignments are not provided in the curriculum.
Students are recommended with additional materials (videos, articles) for self-study and processing.

Course materials and recommended reading

Key literature

1. Laplante P. A., Kassab M. H. Requirements engineering for software and systems. – CRC press, 2022, 395 p.
2. Shishkov B. Designing Enterprise Information Systems: Merging Enterprise Modeling And Software Specification. New York: Springer, 2020. — 242 p.
3. Tsui F., Karam O., Bernal B. Essentials of software engineering. – Jones & Bartlett Learning, 2022., 325 p.
4. Richards M., Ford N. Fundamentals of software architecture: an engineering approach. – O'Reilly Media, 2020, 451 p.
5. Kossiakoff A. et al. Systems engineering principles and practice. – John Wiley & Sons, 2020., 384 p.

Additional literature

1. I Jacobson I., Lawson H. Bud, Ng P.-W., McMahon P, E., Goedicke M. The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons! Association for Computing Machinery and Morgan & Claypool Publishers, 2019. — 402 p
2. A. Bubnov, S. Bubnov, K. Maikov, "Rozrobka i analiz vimog to the program zabezpechennya ", COURSE, 2018, 176 p.
3. Mejia J., Muñoz M., Rocha A., Quiñonez Y. (Eds.). New Perspectives in Software Engineering: Proceedings of the 9th International Conference on Software Process Improvement (CIMPS 2020) Springer, 2021. — 389 p.
4. Rosen C. Guide to Software Systems Development: Connecting Novel Theory and Current Practice. New York: Springer, 2020. — 208 p.
5. Standard for Software Verification and Validation Plans (ANSI / IEEE standard 1012-1986)
6. D'Andrade Brian. Software Engineering: Artificial Intelligence, Compliance, and Security. Nova Science Publishers, 2021. — 262 p.

Assessment and grading

Criteria for assessment of student performance, and the final score structure

100% of the final assessment consists of the results of the assessment in the form of an exam (40%) and the current assessment (60%):
- 8 laboratory works (5% each);
- 2 test papers (10% each).

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by	08.06.2023	Head of the department Ihor HAMAIUN
	08.06.2023	Guarantor of the educational program Uliya LITVINOVA