

	CURRICULUM AL DISCIPLINEI/MODULULUI	Cod:	CD/M – 8.1
	SISTEME SOFTWARE COMPLEXE. PROIECT SEMESTRIAL	Pagina	1/13

APROBAT

la ședința DIIS

nr. ____ din ____ 2024 ____

Șef DIIS

Viorica Sudacevschi, conf. univ., dr.

APROBAT

la ședința Consiliului FCIM

nr. ____ din ____ 2024 ____

Președintele Consiliului FCIM

Dumitru Ciorbă, conf. univ., dr.

Program de master: Managementul aplicațiilor informaționale

Denumirea disciplinei: SISTEME SOFTWARE COMPLEXE. PROIECT SEMESTRIAL

Benefeciari: Studenții anului I, învățământ cu frecvență

Ciclul de învățământ: Studii superioare de Masterat, ciclul II

Numărul de credite ECTS: 10 (80 ore în auditoriu și 220 ore de activități individuale ale studentului, 1 credit = 15 ore de activități în auditoriu și 15 ore de activități individuale ale studentului)

Titularii disciplinei:

conf. univ., dr. Kulev Mihail _____

semnătura titularului de curs

asist. univ. Bucicovschi Evghenii _____

semnătura titularului de curs

	CURRICULUM AL DISCIPLINEI/MODULULUI	Cod:	CD/M – 8.1
	SISTEME SOFTWARE COMPLEXE. PROIECT SEMESTRIAL	Pagina	2/13

I. PRELIMINARII

Sisteme software complexe reprezintă aplicații și sisteme software, în care funcționalitățile acestora sunt organizate și oferite sub formă de servicii independente și complexe, accesibile prin intermediul unor interfețe standardizate. Serviciile sunt componente modulare care îndeplinesc funcții specifice și sunt proiectate pentru a comunica între ele, indiferent de tehnologiile, limbajele de programare sau platformele pe care sunt dezvoltate. Pentru a utiliza cu succes aceste metode și tehnici programatorul trebuie să aibă pregătirea teoretică respectivă și să dispună de deprinderile practice necesare.

Scopul principal al disciplinei „Sisteme software complexe. Proiect semestrial” este studierea metodelor de proiectare a sistemelor software complexe, însușirea metodelor de analiză și sinteză a aplicațiilor și pregătirea studenților să proiecteze, dezvolte și gestioneze sisteme software scalabile, robuste și distribuite, utilizând arhitecturi avansate, tehnologii moderne și principii de securitate, optimizare și integrare, pentru a rezolva provocările aplicative din medii complexe.

Disciplina „Sisteme software complexe. Proiect semestrial” este inclusă în categoria unităților de curs obligatorii de specialitate pentru studenții programului de master *Managementul aplicațiilor informaționale*. Consolidarea materialului teoretic și obținerea abilităților practice se realizează în procesul de îndeplinire a proiectului semestrial.

Obiectivele principale ale disciplinei „Sisteme software complexe. Proiect semestrial” reprezintă formarea la studenți a următoarelor abilități:

- ✓ Cunoașterea și definirea noțiunilor de bază din programare, necesare pentru studierea aprofundată a metodelor de proiectare și dezvoltarea aplicațiilor și sistemelor software complexe;
- ✓ Utilizarea metodelor de optimizare a aplicațiilor complexe;
- ✓ Cunoașterea sistemelor de operare utilizate pentru rularea aplicațiilor cu arhitectură orientată pe servicii și a specificului de functionare a acestora;
- ✓ Utilizarea metodologiilor de bază pentru dezvoltarea aplicațiilor și sistemelor software complexe;
- ✓ Obținerea experienței și îndemnărilor practice în dezvoltarea aplicațiilor și sistemelor software complexe.

Obiectivele specifice ale disciplinei sunt:

- Dezvoltarea abilității de a alege și implementa arhitecturi adecvate pentru sisteme scalabile și distribuite.
- Familiarizarea cu principiile comunicării între componente, sincronizarea proceselor și gestionarea datelor distribuite.
- Aplicarea microserviciilor, DevOps, CI/CD, bazelor de date NoSQL și containerelor pentru a crea soluții eficiente și scalabile.
- Îmbunătățirea performanței aplicațiilor și implementarea măsurilor de protecție împotriva vulnerabilităților.
- Integrarea teoriei și practicii pentru a proiecta, dezvolta, testa și implementa soluții complete, în conformitate cu cerințele specifice.

II. PRECONDIȚII DE ACCES LA DISCIPLINĂ/MODUL

Pentru a atinge obiectivele cursului studenții trebuie să posede competențe și abilități formate la următoarele unități de curs: Limbaj de programare orientate pe obiecte, Analiza și modelarea sistemelor, Baze de date relaționale (SQL) și a celor NoSQL, Sisteme distribuite, Protocoale de comunicare, Testare software, Aplicații și servicii Cloud..

	CURRICULUM AL DISCIPLINEI/MODULULUI		Cod:	CD/M – 8.1
	SISTEME SOFTWARE COMPLEXE. PROIECT SEMESTRIAL		Pagina	3/13

III. COMPETENȚELE CARE URMEAZĂ A FI DEZVOLTATE

Competențele formate de această unitate de curs vor servi ca bază pentru formarea competențelor profesionale în cadrul unităților de curs "Practica de cercetare" și "Susținerea tezei de master".

Disciplina „Sisteme software complexe. Proiect semestrial” prevede formarea următoarelor competențe profesionale și transversale:

CPM1. Sistemul informatic (SI) și alinierea strategiilor de afaceri.

Coordonează procesele de proiectare și implementare pe termen lung a soluțiilor inovative pentru Sisteme Informatic.

CPM5. Proiectarea și dezvoltarea aplicațiilor. .

Acționează creativ pentru a dezvolta aplicații și a selecta opțiunile tehnice adecvate. Participă la alte activități de dezvoltare. Optimizează dezvoltarea, întreținerea și performanța aplicațiilor prin utilizarea modelelor de design și prin reutilizarea soluțiilor testate.

CPM9. Îmbunătățirea proceselor

Realizează și autorizează implementarea de inovații și îmbunătățiri care vor spori competitivitatea și/sau eficiența. Demonstrează conducerii superioare posibile beneficii pentru afaceri care vor putea fi obținute în urma implementării schimbărilor propuse.

CTM1. Autonomie și responsabilitate.

Execută unele sarcini profesionale complexe, în condiții de autonomie și de independență profesională.

CTM2. Interacțiune socială.

Își asumă funcții de conducere în activitățile profesionale sau în structurile organizatorice.

CTM3. Dezvoltare personală și profesională.

Exercită autocontrolul procesului de învățare, previziunea nevoilor de formare, analiza critică a propriei activități profesionale.

IV. ADMINISTRAREA UNITĂȚII DE CURS

Codul disciplinei	Anul predării	Semestru I	Numărul de ore					Evaluarea		
			Prelegeri	Seminare	Lucrări de laborator	Proiect de an	Lucrul individual	Credite	Curentă	Finală
S.O.005	Învățământ cu frecvență									
	II	III				80	220	10		PA

V. REZULTATELE ÎNVĂȚĂRII, CONȚINUTURI ȘI METODE DIDACTICE APLICATE

Rezultatele învățării. Studentul trebuie:	Conținuturi		Metode de predare	Realizarea în timp (ore)*			
	Prelegeri	Seminare / Lucrări de laborator		învățământ cu frecvență		învățământ cu frecvență redușă	
				Prel.	Sem./ lab	Prel.	Sem./ l. lab
1	2	3	4	5	6	7	8
să cunoască: ▪ Obiectivele și scopul disciplinei ▪ Conceptele principale ale disciplinei să fie capabil: ▪ Să aplice conceptele generale ale sistemelor software complexe ▪ Să definească sisteme software complexe	Tema 1. Analiza domeniului și identificarea cerințelor proiectului ▪ Definirea scopului proiectului. ▪ Identificarea cerințelor funcționale și non-funcționale. ▪ Realizarea unui document de cerințe (SRS – Software Requirements Specification). Rezultat: Documentul cerințelor proiectului.	Consultatii la proiectul semestrial 1. Stabilirea temelor proiectului semestrial. Nominalizarea competențelor și atribuțiilor proiectantului. Stabilirea domeniului de cercetare. Cercetarea tematicii în domeniu. Studierea practică și cercetarea domeniului.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0
să cunoască: ▪ Arhitecturi software avansate ▪ Principiile creării unui design de nivel înalt să fie capabil: ▪ Să aplice principiile procesul proiectării aplicațiilor în baza arhitecturii software avansate	Tema 2. Arhitecturi software avansate ▪ Alegerea unei arhitecturi potrivite pentru proiect (monolitic, microservicii, event- driven, etc.). ▪ Crearea unui design de nivel înalt, incluzând	Consultatii la proiectul semestrial 2. Formularea sarcinii proiectului semestrial. Formularea obiectivelor proiectului de an. Formularea scopului proiectului semestrial.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0

Să utilizeze protocoale de comunicare între servicii	diagrame UML (diagrama de componente, de clase, etc.). Rezultat: Specificația arhitecturii proiectului.	Studierea practică procesului de formulare a sarcinilor și a obiectivelor proiectului semestrial.					
să cunoască: - Diferețele între sisteme distribuite și concurente să fie capabil: - Să implementeze și să aplice corect rezultatele teoretice în procesul de proiectare a sistemelor soft complexe	Tema 3. Sisteme distribuite și concurente - Detalierea componentelor și modul în care acestea vor comunica între ele. - Proiectarea mecanismelor pentru gestiunea concurenței și a comunicării în sistem. Rezultat: Diagrame detaliate ale componentelor și specificații de flux.	Consultatii la proiectul semestrial 3. Elaborarea planului de management al proiectului. Formularea cerințelor funcționale și nonfuncționale ale proiectului semestrial. Definirea activităților. Descrierea activităților generale sau particulare. Studierea practică și cercetarea procesului de formulare a cerințelor funcționale și nonfuncționale ale proiectului semestrial.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0
să cunoască: - Configurarea instrumentelor pentru CI/CD - Modalități de utilizare corectă a noțiunilor și rezultatelor fundamentale să fie capabil: - Să implementeze și să aplice corect rezultatele teoretice în procesul de proiectare a	Tema 4. Managementul dependențelor și integrarea continuă - Configurarea instrumentelor pentru CI/CD (Continuous Integration/Continuous Deployment). - Instalarea și configurarea mediului de	Consultatii la proiectul semestrial 4. Planificarea resurselor software și hardware necesare. Identificarea resurselor software și hardware necesare. Identificarea mediilor de dezvoltare IDE. Identificarea SDK-urilor necesare.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0

sistemelor soft complexe	lucru (framework-uri, librării). Integrarea dependențelor esențiale. Rezultat: Mediu de dezvoltare configurat, pipeline de CI/CD activ.	Identificarea categoriilor de sisteme țintă la care se va orienta aplicația. Instalarea sistemelor de operare și softului specializat pentru dezvoltarea aplicațiilor. Studierea practică și cercetarea procesului de identificare și planificare a resurselor software, hardware și de infrastructură necesare.					
să cunoască: - Modalități de utilizare corectă a noțiunilor și rezultatelor fundamentale să fie capabil: - Să implementeze funcționalitățile de bază ale aplicației	Tema 5. Introducere în sisteme software complexe. Dezvoltarea modului de bază al aplicației - Implementarea funcționalităților de bază ale aplicației (MVP – Minimum Viable Product). - Testarea inițială a interacțiunilor între componente. (compoziția serviciilor). Rezultat: MVP funcțional al proiectului.	Consultatii la proiectul semestrial 5. Realizarea configurațiilor necesare. Selectarea și instalarea limbajelor de programare necesare. Instalarea și configurarea emulatoarelor necesare. Instalarea și configurarea plugin-urilor necesare. Studierea practică și cercetarea procesului de instalare și configurare a instrumentelor necesare.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0
să cunoască: - Aspectele de performanță și optimizare - Metodele de proiectare a funcționalităților avansate	Tema 6. Aspecte de performanță și optimizare. Implementarea funcționalităților avansate	Consultatii la proiectul semestrial 6. Elaborarea interfeței de comunicare între servicii. Integrarea serviciilor.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri	0	8/0	0	0/0

să fie capabil: Să aplice metodele avansate de proiectare la dezvoltarea aplicațiilor	- Adăugarea de funcționalități suplimentare. - Optimizarea fluxurilor și îmbunătățirea performanței. Rezultat: Aplicație complet funcțională cu funcționalități avansate.	Studierea practică și cercetarea procesului de elaborare a interfețelor de comunicare între servicii.	- Conversații - Discuții				
să cunoască: - Modalități de utilizare corectă a noțiunilor și rezultatelor fundamentale să fie capabil: - Să implementeze și să aplice corect rezultatele teoretice în procesul de proiectare a sistemelor soft complexe	Tema 7. Managementul datelor în sisteme doftware complexe - Implementarea soluției de stocare a datelor. - Gestionarea datelor distribuite, replicarea și asigurarea consistenței. Rezultat: Aplicație complet funcțională cu funcționalități avansate.	Consultatii la proiectul semestrial 7. Implementarea algoritmilor de funcționare a serviciilor. Elaborarea modulelor de comunicare. Elaborarea modulelor de management al datelor. Elaborarea algoritmilor de funcționare a aplicației. Studierea practică și cercetarea procesului de elaborare și implementare a algoritmilor de funcționare a aplicației.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0
să cunoască: - Modalități de utilizare corectă a noțiunilor și rezultatelor fundamentale - Metodele de testare și verificare să fie capabil: - Să implementeze și să aplice	Tema 8. Testarea și verificarea sistemelor software complexe - Crearea testelor unitare, de integrare și funcționale. - Testarea performanței și identificarea	Consultatii la proiectul semestrial 8. Implementarea serviciilor web și a funcțiilor API pe partea server. Elaborarea funcționalităților complete pe partea server. Elaborarea	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0

corect rezultatele teoretice în procesul de proiectare a sistemelor soft complexe	eventualelor blocaje. Rezultat: Document cu rezultatele testării și soluții pentru remedierea erorilor.	interfeței pentru comunicarea și deservirea clienților. Implementarea serviciilor web. Elaborarea protocoalelor de comunicare. Studierea practică și cercetarea procesului de elaborare și implementare a protocoalelor de comunicare.					
să cunoască: - Modalități de utilizare corectă a noțiunilor și rezultatelor fundamentale - Aspectele de securitate în sisteme software complexe să fie capabil: - Să desfășoare testările aplicației împotriva atacurilor cibernetice comune	Tema 9. Securitatea în sisteme software complexe - Auditarea securității proiectului. - Implementarea măsurilor pentru prevenirea vulnerabilităților - Testarea aplicației împotriva atacurilor cibernetice comune. Rezultat: Aplicație securizată, cu documentație privind măsurile de securitate.	Consultatii la proiectul semestrial 9. Testarea integrității și funcționalității aplicației orientate pe servicii. Testarea modulelor funcționale. Testarea modulului de comunicare. Testarea protocoalelor de comunicare. Testarea algoritmilor de funcționale a sistemelor software complexe. Studierea practică și cercetarea procesului de testare a modulelor sistemelor software complexe.	- Prezentări Power Point - Demonstrații pe tablă - Expuneri - Conversații - Discuții	0	8/0	0	0/0
să cunoască: Să cunoască procesul de integrarea aplicației cu servicii Cloud sau dispozitive IoT	Tema 10. Sisteme software pentru IoT și Cloud Integrarea aplicației cu servicii Cloud sau	Consultatii la proiectul semestrial 10. Adăugarea serviciilor în sistemul realizat în baza	- Prezentări Power Point - Demonstrații pe tablă	0	8/0	0	0/0

să fie capabil:

- Să implementeze și să aplice corect rezultatele teoretice în procesul de proiectare a sistemelor soft complexe

dispozitive IoT, dacă este cazul.

- Pregătirea unui demo pentru prezentarea finală.

Rezultat: Proiectul finalizat și funcțional, gata pentru prezentare.

arhitecturilor orientate pe servicii. Crearea și înregistrarea conturilor de utilizator. Optimizarea sistemelor software complexe. Portarea aplicațiilor pe diferite sisteme de operare. Desfășurarea sistemelor software complexe

Studiarea practică și cercetarea procesului de portare a aplicațiilor pe diferite sisteme de operare, optimizarea sistemelor software complexe

- Expuneri
- Conversații
- Discuții

Total:
0
80/0
0
0/0

 UNIVERSITATEA TEHNICĂ A MOLDOVEI	CURRICULUM AL DISCIPLINEI/MODULULUI	Cod:	CD/M – 8.1
	ARHITECTURI ORIENTATE PE SERVICII	Pagina	10/13

VI. SUGESTII PENTRU ACTIVITATEA INDIVIDUALĂ A STUDENȚILOR

Nr. crt.	Capitol, temă	Conținut activitate individuală	Durata, ore	Forma de control	Termeni de control (în zile)
1	2	3	4	5	6
1	Analiza surselor de date bibliografice disponibile și formularea temei proiectului	Definirea problemei și formularea temei proiectului. Selectarea surselor de date bibliografice și stabilirea tehnologiilor de dezvoltare ale soft-ului.	20	Discuție	2
2	Definirea tematicii proiectului	Formularea problemei. Formularea scopului și a obiectivelor proiectului.	24	Discuție	3
3	Proiectarea sistemului conform tematicii	Perfectarea schemelor de structură a sistemului, argumentarea alegerii tehnologiilor și a librăriilor software utilizate	40	Discuție Prezentare format electronic	10
4	Alegerea și configurarea mediului de dezvoltare și componentelor sistemului	Configurarea instrumentelor software necesare. Selectarea pachetelor și bibliotecilor pentru implementarea proiectului. Selectarea componentelor sistemului. Crearea mediului de testare și rularea testelor inițiale.	20	Discuție Prezentare format electronic	5
5	Asamblarea sistemului	Interconectarea componentelor într-un sistem.	40	Discuție Prezentare prototipului sistemului	10
6	Testarea funcționării sistemului	Depanarea a fiecărui funcționalități componentelor sistemului	40	Discuție Prezentare prototipului sistemului	10
7	Scrierea și prezentarea raportului final a proiectului	Documentarea etapelor implementării, prezentarea rezultatelor obținute și concluziilor. Pregătirea prezentării finale a proiectului, Verificarea standardelor de formatare a raportului.	36	Conversație Prezentare format electronic	10
Total			220		50

	CURRICULUM AL DISCIPLINEI/MODULULUI	Cod:	CD/M – 8.1
	ARHITECTURI ORIENTATE PE SERVICII	Pagina	11/13

VII. UTILIZAREA IA GENERATIVE

Permisuni de utilizare:

Utilizarea IA generative în cadrul temelor și proiectelor este permisă, cu condiția ca studenții să respecte următoarele reguli:

- IA generativă poate fi utilizată pentru generarea de idei, structuri de text sau cod, dar toate materialele generate trebuie să fie revizuite și ajustate de către student pentru a se asigura că acestea corespund cerințelor academice.
- Orice utilizare a IA generative trebuie să fie declarată în secțiunea de appendice a fiecărei lucrări, folosind fraza: "În timpul pregătirii acestei lucrări, autorul a utilizat [NUME INSTRUMENT / SERVICIU] în scopul [MOTIV]. După utilizarea acestui instrument/serviciu, autorul a revizuit și editat conținutul după cum a fost necesar și își asumă întreaga responsabilitate pentru conținutul lucrării."

Restricții de utilizare:

- Studenții nu trebuie să considere IA generativă ca o sursă de încredere pentru informații, deoarece nu oferă referințe clare sau surse documentate.
- Nu este permisă citarea directă a conținutului generat de IA în lucrările academice ca și cum ar fi sursă primară.
- Activitățile în care este interzis utilizarea IA generativă sunt specificare de profesor și sunt de regulă evaluări intermediare și finale sau care nu presupun activități de dezvoltare a competențelor profesionale.

VIII. EVALUAREA DISCIPLINEI

Forma de învățământ	Periodică		Curentă	Proiect/teză	Examen final
	Atestarea 1	Atestarea 2			
Cu frecvență	10%	10%	10%	30%	40%
	Standard minim de performanță				
	Definirea unei probleme actuale și descrierea în ansamblu a soluției/soluțiilor utilizând tehnicile de analiză, proiectare și dezvoltare ale produselor program. Prezența și activitatea la consultațiile proiectului semestrial sunt monitorizate constant, ceea ce permite studenților să îmbunătățească progresul în cadrul proiectului. Evaluarea periodică și curentă este de tip formativ, asigurând un proces continuu de învățare și ajustare a soluțiilor la activitățile de proiectare. Examenul final, fiind o evaluare sumativă, se realizează oral în baza proiectului prezentat public și discuții individuale (în prezența echipei). Aprecierile obținute la examinare sunt individuale și constituie 40% din nota finală.				

IX. CRITERII DE EVALUARE

Denumire	Modul de desfășurare	Pondere pe componente de conținut
		învățământ cu frecvență

	CURRICULUM AL DISCIPLINEI/MODULULUI	Cod:	CD/M – 8.1
	ARHITECTURI ORIENTATE PE SERVICII	Pagina	12/13
Evaluare curentă	Evaluarea curentă a studenților se realizează în baza următoarelor activități: 1) Rezultatele obținute în cadrul proiectului și respectarea etapelor și termenelor stabilite din planul de activitate, îndeplinirea sarcinilor și obiectivelor în intervalele de timp prevăzute. 2) Prezența la activitățile didactice. Criterii de evaluare: corectitudinea îndeplinirii sarcinii proiectului semestrial, rezultatele obținute, ponderea absențelor nemotivate.		10%
Evaluare periodică			
EP 1	Se prezintă realizarea proiectului în volum de 30%. Punctele 1-3 conform tabelului VI (Sugestii pentru activitatea individuală a studenților)		10%
EP 2	Se prezintă realizarea proiectului în mărime de 60%. Punctele 4-6 conform tabelului VI (Sugestii pentru activitatea individuală a studenților)		10%
Proiect/teză	Implementarea integrală a proiectului.		30%
Examen final	Susținerea finală a proiectului, cu prezentare PowerPoint sau similară, discuții la tema proiectului.		40%

X. LISTA DE SUBIECTE PENTRU EVALUĂRI PERIODICE ȘI CEA FINALĂ

CHESTIONAR PENTRU EP 1

Evaluarea periodică 1 se realizează în baza raportului intermediar de proiect, care include aspecte ale lucrului individual și în echipă:

1. Argumentarea tematicii proiectului.
2. Formularea scopului și obiectivelor proiectului.
3. Descrierea componentelor și tehnologiilor utilizate la realizarea proiectului.
4. Descrierea și argumentarea metodologiilor utilizate la dezvoltarea proiectului.

CHESTIONAR PENTRU EP 2

Evaluarea periodică 2 se realizează în baza raportului intermediar de proiect, care include aspecte ale lucrului individual și în echipă:

1. Descrierea proiectului propriu zis.
2. Agenda activităților realizate pentru atingerea obiectivelor.
3. Analiza bibliografiei.
4. Perspective de dezvoltare ale sistemului proiectat.

CHESTIONAR PENTRU EXAMEN

Examinarea finală se realizează în baza raportului de proiect, care include aspecte ale lucrului individual și în echipă:

5. Argumentarea tematicii proiectului.
6. Formularea sarcinii și obiectivelor proiectului.
7. Elaborarea schemei de structură a sistemului.
8. Descrierea componentelor și tehnologiilor utilizate la realizarea proiectului.
9. Descrierea și argumentarea metodologiilor utilizate la dezvoltarea proiectului.
10. Descrierea proiectului propriu zis.
11. Agenda activităților realizate pentru atingerea obiectivelor.
12. Analiza bibliografiei.

	CURRICULUM AL DISCIPLINEI/MODULULUI	Cod:	CD/M – 8.1
	ARHITECTURI ORIENTATE PE SERVICII	Pagina	13/13

13. Perspective de dezvoltare ale sistemului proiectat.
14. Concluzii.

XI. REFERINȚE BIBLIOGRAFICE

Obligatorii

1. Dorin Bocu, R. Bocu, Întroducere în universul ingineriei softului : Despre abordarea metodologică în industria softului: Întroducere în modelare. Perspectiva UML, București MATRIX ROM 2023 ISBN: 978-606-25-0816-6
2. Дж. Рамбо, М. Блаха, UML 2.0 : Объектно-ориентированное моделирование и разработка, Москва | Санкт-Петербург | Нижний Новгород Питер 2007, ISBN: 5- 469-00814-2
3. Software Engineering, tutorial online, <https://www.infobooks.org/pdfview/14007-software-engineering-tutorialspoint/>
4. H. Sayama, Introduction to the Modeling and Analysis of Complex Systems. Available: <https://archive.org/details/introduction-to-the-modeling-and-analysis-of-complex-systems>.
5. S. Matthews, T. Newhall, and K. C. Webb, Dive into Systems. Available: <https://diveintosystems.org>.
6. Defense Acquisition University, Systems Engineering Fundamentals. Available: <https://www.infobooks.org/free-pdf-books/engineering/systems-engineering/>.
7. Infobooks.org, Software Engineering. Available: <https://www.infobooks.org/free-pdf-books/computers/software-engineering/>.
8. Universitatea Politehnica Timișoara, "Proiectarea Sistemelor Software Complexe – Curs 1: Introducere". Available: <https://www.aut.upt.ro/staff/diercan/data/PSSC/curs-01.pdf>.

Suplimentare

1. M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol, CA, USA: O'Reilly Media, 2017.
2. L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice. 3rd ed., Boston, MA, USA: Addison-Wesley, 2012.
3. S. Newman, Building Microservices: Designing Fine-Grained Systems. 2nd ed., Sebastopol, CA, USA: O'Reilly Media, 2021.
4. R. C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston, MA, USA: Prentice Hall, 2017.
5. A. S. Tanenbaum and M. Van Steen, Distributed Systems: Principles and Paradigms. 2nd ed., Upper Saddle River, NJ, USA: Pearson, 2007.