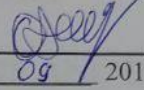


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНИЙ
УНІВЕРСИТЕТ

Кафедра комп'ютерних технологій і мехатроніки

«ЗАТВЕРДЖУЮ»

Гарант освітньо-професійної програми
«Програмне забезпечення систем»
першого (бакалаврського) рівня вищої
освіти, завідувач кафедри КТМ, д.т.н.,

професор  Ніконов О.Я.
« 03 » 09 2019р.

СИЛАБУС
АРХІТЕКТУРА ТА АНАЛІЗ ВИМОГ ДО
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ /
ARCHITECTURE AND ANALYSIS OF SOFTWARE
REQUIREMENTS
SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Харків 2019

Автор: Шапошнікова Олена Павлівна, доцент кафедри комп'ютерних технологій і мехатроніки

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних технологій і мехатроніки, протокол № 18 від «27» червня 2019 р.

СИЛАБУС

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ /

SOFTWARE ARCHITECTURE AND DESIGN SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Анотація курсу

1. Викладачі

1.1. Лектор: Шапошнікова Олена Павлівна

- доцент кафедри комп'ютерних технологій та мехатроніки;
- педагогічний стаж – 25 років
- контактний телефон +38-057-707-37-43
- e-mail: shaposhnikovaep@gmail.com
- наукові інтереси: архітектура ПЗ, аналіз вимог до ПЗ, управління ІТ проектами, якість ПЗ та тестування.

1.2. Асистент лектора:

2. Дисципліна «Архітектура та аналіз вимог до програмного забезпечення»

- рік навчання: 2;
- семестр навчання: 4;
- кількість годин за семестр: 90, в т.ч.
 - лекційних: 16;
 - практичних занять: 32;
 - на самостійне опрацювання: 42;
- кількість аудиторних годин на тиждень
 - лекційних: 2 (раз на два тижні);
 - практичних занять: 2.

3. Час та місце проведення

- аудиторні заняття – відповідно до розкладу ХНАДУ, ауд. 214, 216;
- позааудиторна робота – самостійна робота студента із використанням технологій віртуалізації MS Visio, Visual Use case.

4. Пререквізити та постреквізити навчальної дисципліни:

- **перереквізити:** «Основи інформаційних технологій», «Алгоритмізація та програмування», «Операційні системи», «Об'єктно-орієнтоване програмування», «Алгоритми і структури даних».
- **постреквізити:** «Програмування баз даних», «Web-програмування», «Управління ІТ проектами», «Якість програмного забезпечення та тестування».

5. Характеристика дисципліни:

5.1. Призначення навчальної дисципліни: вивчення дисципліни «Архітектура та проектування програмного забезпечення» займає вагомe місце у структурі отримання знань за освітньою програмою «Програмне забезпечення систем». Процес проектування програмного забезпечення (ПЗ) починається із вивчення предметної області та формулювання концепції проекту, визначення його границь, формулювання вимог до ПЗ. Наступним етапом є процес розроблення архітектури, який дозволяє знизити складність системи шляхом абстракції та розмежування повноважень. Архітектура ПЗ об'єднує різні точки зору на систему, що є вагомим аргументом щодо необхідності та доцільності розроблення архітектури ПЗ. Ця дисципліна формує системний підхід до проектування ПЗ та інженерний світогляд фахівця у галузі ІТ.

5.2. Мета вивчення дисципліни: «Архітектура та проектування програмного забезпечення» є теоретична та практична підготовка студентів, яка має забезпечити отримання ними основних знань у галузі сучасних технологій проектування, інженерії вимог до програмного забезпечення (ПЗ) та вивчення архітектури програмного забезпечення, отримання практичних навичок реалізації програмних систем, основи моделювання і аналізу програмних систем, аналізу розробки, специфікації та управління вимогами.

5.3. Задачі вивчення дисципліни: теоретична та практична підготовка майбутніх фахівців з питань:

- Етапи та фази життєвого циклу програмного продукту;
- формальні моделі систем; моделі предметних областей інформаційних систем; стандарти IDEF1, IDEF3, IDEF5;
- методи аналізу та синтезу інформаційних систем;
- аналіз структур інформаційних систем; механізми інтеграції систем, методології моделювання структури та поведінки програмних систем з використанням мови UML.
- розроблення моделі предметних областей логічної та фізичної архітектури програмної системи, схеми послідовностей, станів та взаємодій компонентів програмної системи;
- проведення досліджень характеристик компонентів та інформаційних систем в цілому;
- застосовування на практиці методів та засобів проектування інформаційних систем;
- аналіз інформаційних систем;
- розроблення математичних моделей інформаційних систем;
- застосовування уніфікованої мови моделювання для побудови діаграм,
- уміння ставити задачу та розробляти прикладну програмну систему з використанням сучасних технологій розробки ПЗ.

5.4. Зміст навчальної дисципліни: відповідає навчальній та робочій програмі, яка відповідає запитам роботодавців.

5.5. План вивчення дисципліни

Результати навчання	Навчальна діяльність	Робочий час студента (год.)	Оцінювання (бал)
1	2	3	4
Тема 1. Введення в дисципліну. Основні поняття та визначення.		11	11
Знати поняття архітектури ПЗ, архітектурні зразки, еталонні моделі та варіанти архітектур, архітектурні структури та	Лекція №1. Введення в дисципліну. Основні поняття та визначення. План лекції: 1. Поняття архітектури ПЗ. 2. Архітектурні зразки, еталонні моделі та варіанти архітектур.	2	

представлення, відношення між структурами. Вміти виявляти цілі діяльності замовника, визначати набір задач, які необхідно вирішувати, визначати набір сутностей з якими треба мати справу при вирішенні задач. SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	3. Архітектурні структури та представлення. 4. Відношення між структурами. Список рекомендованих джерел: Основний: 1-5 Додатковий: 1-3 Інтернет-ресурси: 1-3		
	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: 1. Поняття про предметну область проекту. 2. Формальні визначення. 3. Класифікація моделей.	5	
	Практичне заняття № 1. «Аналіз предметної області. Визначення концепції проекту» Мета: Отримати навички виявлення реальних потреб замовника та з'ясування сенсу висловлених вимог, а також розроблення концепції проекту. Завдання: 1. Ознайомитися з видами моделей. 2. Ознайомитися із методологіями IDEF. 3. Вивчити синтаксис IDEF0-моделей. План заняття: –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 2. Варіанти архітектур програмних систем.		11	13
Знати архітектури, засновані на рівнях абстракцій, на портах, на потоках даних, архітектури незалежних компонентів, сервіс-орієнтовані архітектури. Вміти розуміти мову, якою розмовляють	Лекція 2. Варіанти архітектур програмних систем. План лекції: 1. Архітектура, заснована на рівнях абстракцій. 2. Архітектура, заснована на портах. 3. Архітектура, заснована на потоках даних. 4. Архітектура незалежних компонентів. 5. Сервіс-орієнтовані архітектури.	2	

замовники; розробляти контекстну діаграму та виконувати її декомпозицію, створювати діаграму дерева вузлів. SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	Список рекомендованих джерел: Основний: 1-7, 9 Додатковий: 1-2 Інтернет-ресурси: 1-4 Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Case-засоби для розроблення вимог до ПЗ. 1. Поняття про CASE-засоби для розроблення вимог; 2. Необхідні та достатні функції CASE-засобу для розроблення вимог; 3. Виконати аналітичний огляд CASE-засобів;		
	Практичне заняття № 2. Побудова функціональної моделі проекту та її декомпозиція. <i>Мета:</i> навчитися створювати модель системи в стандарті IDEF0. <i>Завдання:</i> 1. Розробити контекстну діаграму. 2. Створити діаграму декомпозиції. 3. Створити діаграму дерева вузлів. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	5	4
Тема 3. Життєвий цикл ПЗ.		11	13
Знати про життєвий цикл ПЗ, основні, допоміжні та організаційні процеси ЖЦ ПЗ, взаємозв'язок між процесами ЖЦ ПЗ, види моделей і стадії ЖЦ ПЗ.	Лекція 3. Життєвий цикл ПЗ. <i>План лекції:</i> 1. Поняття життєвого циклу (ЖЦ) ПЗ. 2. Основні, допоміжні та організаційні процеси ЖЦ ПЗ. 3. Взаємозв'язок між процесами ЖЦ ПЗ. 4. Види моделей і стадії ЖЦ ПЗ. Список рекомендованих джерел: Основний: 2, 4, 5 Додатковий: 2, 8 Інтернет-ресурси: 1-4	2	
Вміти визначати цільову функцію, основні функції та сервісні функції, формувати дерево функцій.	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного	5	

SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Методологія BPMN. 1. Поняття методології BPMN; 2. Синтаксис BPMN; 3. Ознаки класифікації подій.		
	Практичне заняття № 3. Розроблення дерева функцій проекту та вимог до проекту. <i>Мета: навчитися аналізувати і синтезувати систему за допомогою дерева функцій</i> <i>Завдання:</i> 1. Визначити цільову функцію; 2. Визначити основні функції; 3. Визначити сервісні функції; 4. Сформулювати дерево функцій. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 4. Проектування ПЗ. Визначення вимог і цілей програмного продукту.		11	11
Знати процес проектування, як послідовної трансляції вимог, які пред'являються до ПЗ, методологію вирішення задач проектування по ПЗ по Г. Майерсу, рівні вимог до ПЗ. Вміти Моделювати поведінку системи на логічному рівні шляхом створення діаграм станів.	Лекція 4. Проектування ПЗ. Визначення вимог і цілей програмного продукту. <i>План лекції:</i> 1. Процес проектування, як послідовна трансляція вимог, які пред'являються до ПЗ. 2. Методологія вирішення задач проектування по ПЗ по Г. Майерсу. 3. Рівні вимог до ПЗ. Список рекомендованих джерел: Основний: 4, 5, 7, 14 Додатковий: 4, 5 Інтернет-ресурси: 8, 9	2	
SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики,	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання:	5	

колективний тайм менеджмент, знання іноземної мови.	Еволюція платформних архітектур. 1. Автономна архітектура, 2. Централізована архітектура 3. Розподільна архітектура: – Архітектура «файл-сервер», – Архітектура «клієнт-сервер», – Архітектура веб-додатків.		
	Практичне заняття № 4. Моделювання поведінки системи на логічному рівні (розроблення діаграми станів) <i>Мета: навчитися аналізувати поведінку об'єкта в декількох прецедентах.</i> <i>Завдання:</i> 1. Визначити клас проекту для якого буде розроблена діаграма станів, 2. Вивчити основний набір символів UML для створення діаграми. 3. Розробити діаграму. 4. Проаналізувати стани та способи реакції на події. <i>План заняття:</i> – Актуалізація теоретичного матеріалу – Виконання завдань практичної роботи – Презентація виконаної роботи	4	
Тема 5. Проектування архітектури ПЗ. Методологія проектування. Модульність.		11	12
Знати методологію проектування, модулі, модульно-інтерфейсний підхід, обґрунтування модульності, внутрішню характеристику модуля – зв'язність, зціплення модулів – зовнішня характеристика модуля. Вміти Моделювати взаємодію об'єктів програмної системи в часі та також обмін повідомленнями	Лекція 5. Проектування архітектури ПЗ. Методологія проектування. Модульність. <i>План лекції:</i> 1. Модулі, модульно-інтерфейсний підхід. 2. Обґрунтування модульності. 3. Внутрішня характеристика модуля – зв'язність. 4. Зціплення модулів – зовнішня характеристика модуля. Список рекомендованих джерел: Основний: 7-11 Додатковий: 4-7 Інтернет-ресурси: 10, 11	2	
	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у	5	

між об'єктами шляхом створення діаграми послідовності. SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Розподілені архітектури. 1. Класифікація програмних систем. 2. Основні ознаки розподілених систем. 3. Недоліки розподілених систем. 4. Проблеми проектування розподілених систем.		
	Практичне заняття № 5. Моделювання поведінки програмної системи на логічному рівні (розроблення діаграми послідовності). <i>Мета: навчитися моделювати взаємодію об'єктів системи в часі, а також обмін повідомленнями між ними.</i> <i>Завдання:</i> 1. Вивчити основний набір символів для створення діаграми. 2. Визначити поняття Об'єкт і Повідомлення. 3. Розробити діаграму. 4. Проаналізувати отриману діаграму. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 6. Багатошарова архітектура ПЗ		11	14
Знати Загальні принципи проектування 3 використанням багатошарової програмної архітектури, шаблони проектування багатошарової архітектури. Вміти Створювати специфікації об'єктів окремих класів, які взаємодіють 3	Лекція 6. Багатошарова архітектура ПЗ <i>План лекції:</i> 1. Логічна структура багатошарової архітектури. 2. Поняття рівнів та слоїв. 3. Шаблони проектування багатошарової архітектури. 4. Проектування слоїв. Список рекомендованих джерел: Основний: 7-14, 16 Додатковий: 4, 5, 7 Інтернет-ресурси: 7, 12, 13 Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних	2	
		5	

метою реалізації окремих варіантів використання в загальному контексті ПК шляхом розроблення діаграми кооперації. SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Архітектура Web-додатків 1. Складові частини Web-додатку. 2. Архітектура багаторівневого Web-додатку. 3. Архітектура змішаного Web-додатку. 4. Архітектура багаторівневого Web-додатку на основі технології CORBA. 5. Архітектура Web-додатків з інтерфейсами OLE DB, ADO и ODBC.		
	Практичне заняття № 6. Моделювання поведінки ІС на логічному рівні (розроблення діаграми кооперації) <i>Мета: навчитися створювати специфікації об'єктів окремих класів, які взаємодіють з метою реалізації окремих варіантів використання в загальному контексті ПК шляхом розроблення діаграми кооперації.</i> <i>Завдання:</i> 1. Вивчити основний набір символів для створення діаграми. 2. Визначити структурні елементи діаграми кооперації. 3. Розробити діаграму. 4. Проаналізувати отриману діаграму. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 7. Паттерни проектування		16	13
Знати Поняття паттерну. Доцільність використання паттернів, Класифікація паттернів: ідіоми, архітектурні паттерни, класифікація за	Лекція 7. Паттерни проектування <i>План лекції:</i> 1. Поняття паттерна проектування. 2. Види паттернів. 3. Породжуючі шаблони. 4. Структурні шаблони. 5. Поведінкові шаблони. Список рекомендованих джерел: Основний: 7-13, 16 Додатковий: 4-7	2	

призначенням. Породжуючі, поведінкові та структурні шаблони. Вміти Розділяти систему на компоненти та представляти загальну конфігурацію і топологію ПС шляхом побудови діаграм компонентів та розгортання.	Інтернет-ресурси: 14-17 Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Сокети. 1. Поняття сокету. 2. Принципи сокетів. 3. Функції сокетів.	6	
SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	Практичне заняття № 7. Розроблення моделі фізичного представлення ІС (розроблення діаграми компонентів та розгортання) Мета: навчитися розділяти систему на компоненти та представляти загальну конфігурацію і топологію ПС. Завдання: 1. Вивчити основний набір символів для створення діаграм. 2. Визначити з яких фізичних часток складається ПС. 3. Визначити типи користувачів. 4. Визначити апаратні та технічні типи пристроїв для виконання системою своїх функцій. 5. Визначити види та потрібну пропускну здатність каналів зв'язку. 6. Розробити діаграми. План заняття: –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	8	
Тема 8. Прийоми об'єктно-орієнтованого проектування.		8	13
Знати Принципи об'єктно-орієнтованої архітектури та її переваги. Концепції та елементи предметно-орієнтованого проектування. Вміти	Лекція 8. Прийоми об'єктно-орієнтованого проектування План лекції: 1. Головні принципи об'єктно-орієнтованої архітектури (OOA). 2. Переваги OOA. 3. Концепції та елементи DDD. Список рекомендованих джерел: Основний: 7-14, 16 Додатковий: 4-7 Інтернет-ресурси: 18	2	

SoftSkills: Комунікативні навички, робота в команді, творчі навички, сприйняття конструктивної критики, колективний тайм менеджмент, знання іноземної мови.	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: СОМ-технологія 1. Поняття СОМ-технології. 2. Проблеми, які вирішуються за допомогою СОМ-технології.	6	
Разом		90 годин/ 3 кредити	100 балів
Підсумковий контроль		Залік	

Рекомендовані джерела інформації

1. Базова література

1. Вигерс Карл, Битти Джой Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. — 736 стр.: ил.
2. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. — М.: Финансы и статистика, 1998. — 176 с.
3. Марка, Д.А. Методология структурного анализа и проектирования SADT / Д.А. Марка, К. МакГоуэн. — М.: МетаТехнология, 1993. — 243 с.
4. Калянов, Г.Н. CASE. Структурный системный анализ (автоматизация и применение) / Г.Н. Калянов. — М.: Лори, 1996. — с.
5. Баркер, Р. CASE*Method. Моделирование взаимосвязей между сущностями / Р. Баркер. — М., 1992. — 233 с.
6. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. — М.: ДИАЛОГ-МИФИ, 2005. — 432 с.
7. UML спецификация. — www.omg.com.
8. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч. — М.: Бином, 2001. — 560 с.
9. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Якобсон. - СПб.: Питер, 2004. - 432 с.
10. Леоненков, А.В. Самоучитель UML 2 / А.В. Леоненков. — СПб.: БХВ - Петербург, 2007. — 576с.
11. Леоненков, А.В. Объектно-ориентированный анализ и проектирование с использованием UML / А.В. Леоненков. — www.intuit.ru.
12. Якобсон, А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо. - СПб.: Питер, 2002. - 496 с.
13. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер. — М.: Издательский дом «Вильямс», 2004. — 544 с.
14. Фаулер, М. UML. Основы. Третье издание. / М. Фаулер. — М.: Символ-Глос, 2006. — 192 с.

15. Анисимов, В.В. Проектирование информационных систем. Часть 1. Структурный подход: конспект лекций / В.В. Анисимов. – Хабаровск: Изд-во ДВГУПС, 2005. – 112 с.

16. Анисимов, В.В. Проектирование информационных систем. Часть 2. Объектно-ориентированный подход: конспект лекций / В.В. Анисимов. – Хабаровск: Изд-во ДВГУПС, 2007. – 100 с.

2. Допоміжна література

1. Орлов, С.А. Технологии разработки программного обеспечения: учеб. / С.А. Орлов. – СПб.: Питер, 2002. – 464 с.

2. Маклаков, С.В. BPwin и ERwin. CASE-средства разработки информационных систем / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2001. – 304 с.

3. Крачтен, Ф. Введение в Rational Unified Process / Ф. Крачтен. – М.: Издательский дом «Вильямс», 2002. – 240 с.

4. Элиенс, А. Принципы объектно-ориентированной разработки программ / А. Элиенс. – М.: Издательский дом «Вильямс», 2002. – 496 с.

5. Ларман, К. Применение UML и шаблонов проектирования: Уч. Пос / К. Ларман. – М.: Издательский дом «Вильямс», 2001. – 496 с.

6. Гранд, М. Шаблоны проектирования в Java / М. Гранд. – М.: Новое знание, 2004. – 559 с.

7. Йордан, Э. Объектно-ориентированный анализ и проектирование систем / Э. Йордан, С. Аргила. – М.: Издательство «ЛЮРИ», 2007. – 264 с.

8. (BPwin) Где? Зачем? Как? / В.И. Дубейковский. – М.: ДИАЛОГ-МИФИ, 2004. – 464 с.

3. Интернет-ресурсы

1. Академия Microsoft: Анализ требований к автоматизированным информационным системам: <http://www.intuit.ru/studies/courses/2188/174/info>

2. АНАЛИЗ ТРЕБОВАНИЙ К ИНФОРМАЦИОННЫМ СИСТЕМАМ Конспект лекций Маглинец Ю.А. <http://ivan-shamaev.ru/wp-content/uploads/2013/06/Information-systems-analysis-and-requirements-analysis.pdf>

3. Анализ требований по Вигерсу (2004). Этапы сбора требований. <http://iiba.ru/requirements-analysis/analysis-of-requirements-wiegers-2004/> Вигерс К. Разработка требований к программному обеспечению <http://www.twirpx.com/file/1073169/>

4. С.И. Клевцов АНАЛИЗ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ СБОРА И ОБРАБОТКИ ДАННЫХ Учебное пособие - http://rtf.sfedu.ru/!mps/umk/strdsgn_ch1.pdf

5. Проектирование микрослужбы, ориентированной на DDD <https://docs.microsoft.com/ru-ru/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>

6. Снова о разработке на основе предметной области (Domain-Driven Design, DDD) - <https://habr.com/ru/post/258693/>

7. Проектирование информационных систем - https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema8/tema8_4

8. Основные термины и определения: <https://studopedia.info/10-35578.html>

9. Диаграмма состояний (диаграмма автомата) UML:
<https://planerka.info/item/diagrammy-sostoyanij-diagrammy-avtomata-uml/>
10. Архитектура распределенных систем: <https://helpiks.org/4-83847.html>
11. Архитектурные стили проектирования:
https://studref.com/320288/informatika/arhitekturnye_stili_proektirovaniya
12. Многослойная архитектура:
<https://studfile.net/preview/6413815/page:13/>
13. Основы архитектуры Web-приложений:
<http://studepedia.org/index.php?vol=1&post=53643>
14. Паттерны проектирования: <https://refactoring.guru/ru/design-patterns>
15. Диаграммы развертывания:
https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema15/tema15_3
16. Диаграммы компонентов
https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema15/tema15_2#p153
17. Програмирование сокетов в Linux:
<http://rsdn.org/article/unix/sockets.xml>
18. Технология COM:
<http://techn.sstu.ru/kafedri/%D0%BF%D0%BE%D0%B4%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F/1/MetMat/murashev/com/lec/>