

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

М. Ю. Карпенко

ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ В УПРАВЛІННІ
ОРГАНІЗАЦІЄЮ

КОНСПЕКТ ЛЕКЦІЙ
Частина 1 (ЗМ 1)

*(для здобувачів другого (магістерського) рівня вищої освіти денної і заочної
форм навчання зі спеціальностей 073 – Менеджмент
і 071 – Облік і оподаткування)*

Харків
ХНУМГ ім. О. М. Бекетова
2024

УДК 004.9(075.8)

Карпенко М. Ю. Інформаційні системи і технології в управлінні організацією. Частина 1 (ЗМ 1) : конспект лекцій для здобувачів другого (магістерського) рівня вищої освіти денної і заочної форм навчання зі спеціальностей 073 – Менеджмент і 071 – Облік і оподаткування / М. Ю. Карпенко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2024. – 46 с.

Автор

канд. техн. наук, доц. М. Ю. Карпенко

Рецензенти:

О. М. Грицунов, доктор фізико-математичних наук, професор кафедри мікроелектроніки електронних приладів та пристроїв Харківського національного університету радіоелектроніки;

О. Б. Костенко, кандидат фізико-математичних наук, доцент кафедри прикладної математики і інформаційних технологій Харківського національного університету міського господарства імені О. М. Бекетова

Рекомендовано кафедрою комп'ютерних наук та інформаційних технологій, протокол № 1 від 31.08.2023

Конспект лекцій складено з метою допомогти студентам спеціальностей 073 – Менеджмент і 071 – Облік і оподаткування при підготовці до занять, заліків та іспитів з дисципліни «Інформаційні системи і технології в управлінні організацією».

© М. Ю. Карпенко, 2024

© ХНУМГ ім. О. М. Бекетова, 2024

ЗМІСТ

ЗМ1 ТИПОЛОГІЯ, ПРИНЦИПИ ПРОЄКТУВАННЯ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	5
1 ТИПОЛОГІЯ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	5
1.1 Значення та напрями розвитку інформаційних систем	5
1.2 Головні поняття інформаційних технологій і систем.....	6
1.2.1 Визначення поняття «інформаційна технологія».....	6
1.2.2 Визначення та різновиди інформації.....	7
1.3 Поняття інформаційної системи та класифікація інформаційних систем..	9
1.3.1 Класифікація за рівнем або сферою діяльності.....	9
1.3.2 Класифікація за рівнем автоматизації процесів управління.....	10
1.3.3 Класифікація за ступенем централізації та інтеграції обробки даних.	12
1.3.4 Класифікація за типом інформаційної системи	13
1.3.5 Класифікація за сферою застосування	13
1.3.6 Класифікація за способом організації	15
1.3.7 Вимоги до інформаційних систем	20
1.4 Питання для самоконтролю.....	22
2 ПРИНЦИПИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ	24
2.1 Основні фази проектування інформаційної системи.....	24
2.1.1 Концептуальна фаза (формування концепції)	25
2.1.2 Підготування технічного завдання	25
2.1.3 Проектування	26
2.1.4 Розроблення.....	26
2.1.5 Уведення системи в експлуатацію.....	26
2.2 Процеси, що відбуваються під час життєвого циклу інформаційної системи	27
2.2.1 Головні процеси життєвого циклу.....	28
2.2.2 Допоміжні процеси життєвого циклу.....	29
2.2.3 Організаційні процеси.....	30
2.3 Структура життєвого циклу інформаційної системи	30
2.3.1 Початкова стадія	31
2.3.2 Стадія уточнення	32
2.3.3 Стадія конструювання.....	32

2.3.4 Стадія передавання до експлуатації	32
2.4 Моделі життєвого циклу інформаційної системи	32
2.4.1 Каскадна модель життєвого циклу інформаційної системи	33
2.4.2 Спіральна модель життєвого циклу	38
2.4.3 Ітераційний підхід до моделі життєвого циклу	42
2.5 Питання для самоконтролю	43
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	45

ЗМІ ТИПОЛОГІЯ, ПРИНЦИПИ ПРОЄКТУВАННЯ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

1 ТИПОЛОГІЯ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Значення та напрями розвитку інформаційних систем

Сьогодні інформаційні ресурси є не менш суттєвими, ніж трудові, матеріальні або енергетичні. Роль інформації настільки значна, що ми можемо говорити про інформаційну економіку (тобто таку, що ґрунтується на інформації) та інформаційну сферу управління (керівників, учених, спеціалістів і службовців тощо). У такому аспекті управління економікою подібне до управління живим організмом, де головну функцію виконує нервова система.

Функцію такої «нервової системи» в економіці здійснюють інформаційні потоки. Будь-яке порушення в потоках інформації обов'язково буде даватися взнаки у вигляді перебоїв у роботі конкретного підприємства або навіть на рівні всього господарства. Без достовірної, повної та своєчасно отриманої інформації неможливо керувати будь-яким виробництвом, своєчасно реагувати на виклики зовнішнього середовища. Саме тому на сьогодні проблема збирання та обробки інформації є одним з ключових завдань управління, вирішити яке можна тільки за допомогою використання сучасної обчислювальної техніки та новітніх інформаційних технологій. Варто зазначити, що сама обчислювальна техніка все ширше використовується як в управлінні виробничими процесами, так і на рівні економічної системи загалом. Однак матеріальні витрати на зберігання, передавання та перероблення інформації все частіше перевищують аналогічні витрати, наприклад, на енергетику. І в цьому контексті проблема інформатизації набуває особливого значення не тільки як технічний, але і як економічний складник.

Принциповим є той факт, що комп'ютеризація суспільства – це не тільки техніка, а й люди. Нова комп'ютерна ідеологія повинна включати не тільки технічні питання, а й підготовку відповідних спеціалістів і керівників.

Керівник, що не володіє комп'ютером, не може сподіватися на прогрес у своїй роботі. Таким чином, комп'ютеризація у загальному розумінні потребує комплексного підходу щодо вирішення широкого спектру проблем – від суто технічних до освітніх та соціальних. Саме для вирішення таких завдань в Україні було створено Національне агентство з питань інформатизації при Президентові України та затверджено Закони «Про Національну програму інформатизації» (74/98-ВР від 04.02.1998) і «Про концепцію Національної програми інформатизації» (75/98-ВР від 04.02.1998).

1.2 Головні поняття інформаційних технологій і систем

1.2.1 Визначення поняття «інформаційна технологія»

Комп'ютеризація будь-якого бізнес-процесу базується на інформаційних технологіях (далі – ІТ). У широкому сенсі ІТ – це загальний термін, що позначає різні технології оброблення та передачі інформації [1–5]. Зазвичай під ІТ розуміють комп'ютерні технології, а саме методи застосування обчислювальної техніки (далі – ОТ) і комп'ютерного програмного забезпечення (далі – ПО) під час роботи з інформацією (виконання функцій: збирання, зберігання, захисту, обробки, передавання й використання даних). До ІТ належать, зокрема, телекомунікація і мікроелектроніка.

Згідно з визначенням ЮНЕСКО, інформаційна технологія – це комплекс взаємопов'язаних наукових і інженерних дисциплін, що вивчають ОТ, способи взаємодії її з людьми та виробничим обладнанням, способи ефективної організації праці людей, а також пов'язані з ними соціальні, економічні й культурні проблеми.

За визначенням ІТАА – Американської асоціації з інформаційних технологій (інформаційні технології асоціації Америки), ІТ – це вивчення, проектування, розроблення, упровадження, підтримка або управління комп'ютерними інформаційними системами, включаючи програмні додатки і комп'ютерне апаратне забезпечення. Бурхливий розвиток галузі ІТ припадає на

кінець 1990-х років і пов'язаний із появою та розвитком інформаційно-комунікаційних технологій.

Деякі автори подають таке визначення терміна «інформаційна технологія»: цілеспрямована організована сукупність інформаційних процесів з використанням засобів обчислювальної техніки, що забезпечують високу швидкість обробки даних, пошуку інформації, розосередження даних і доступ до джерел інформації незалежно від місця їхнього розташування.

Незважаючи на певну відмінність у визначеннях, будь-яка інформаційна технологія спрямована на вирішення завдань за трьома основними напрямками, а саме:

- 1) персоналізація розрахунків;
- 2) використання баз даних, експертних систем і баз знань;
- 3) застосування мереж передавання даних.

Варто зауважити, що останнім часом набув популярності відносно новий термін – «інформаційно-комунікаційні технології» (далі – ІКТ). Під ІКТ розуміють усі технічні засоби обробки та обміну інформацією [3–5] в поєднанні з відповідним математичним та програмним забезпеченням. Крім того, цей термін формально включає «доцифрові» технології, зокрема паперові. Також поняття ІКТ використовують для опису цифрових технологій, зокрема способів комунікації (комунікаційні протоколи, комунікаційне обладнання, медіа), способів зберігання та обробки інформації (обчислення, накопичення даних) тощо. Побутує думка, що термін «інформаційно-комунікаційні технології» став популярним внаслідок поєднання ІТ та телекомунікаційних технологій.

1.2.2 Визначення та різновиди інформації

Інформація (з лат. *«informatio»* – «навчання», «зведення», «сповіщення») – це відомості, які передаються системою знаків будь-якого виду. Поняття «інформація» прийнято визначати за її властивостями, тобто інформація – це явище, яке характеризується наявністю джерела, приймача,

каналу зв'язку тощо [1, 2]. До того ж є безліч властивостей інформації, що досі не вивчені науковцями. Зазначене вище суттєво ускладнює формулювання точного визначення поняття «інформація».

Інформація, зафіксована на матеріальних носіях і зберігається в ІС (бібліотеках, архівах, сховищах, фондах, банках даних, системах знань тощо), називається інформаційними ресурсами.

Інформацію умовно можна розділити на групи за сферою використання [1–3]. Наприклад, економічна інформація – це сукупність відомостей про соціально-економічні процеси, що слугують для управління ними й колективами людей. Характеристики економічної інформації: великі обсяги, багаторазове повторення циклів її отримання і перетворення у встановлені періоди (місяць, квартал, рік тощо), різноманіття джерел і споживачів, значна питома вага рутинних процедур під час її обробки.

Інформація в ІС може бути структурованою або неструктурованою та умовно поділяється на дані та відомості.

Структуровані дані – це інформація у вигляді чисел і тексту, що зберігається в нормалізованих БД. З такими даними можна виконувати різні операції.

До неструктурованої інформації можна віднести, наприклад, офісні електронні документи у форматі *Word* або *Excel*, *PDF*, а також рисунки, креслення, графіки, скановані зображення, повідомлення електронної пошти, вебсторінки, відео та іншу інформацію в електронному вигляді.

На відміну від структурованих даних (які зазвичай розміщуються в середовищі реляційної або багатовимірної СУБД), для зберігання неструктурованої інформації використовують файлову систему ОС або об'єктно-орієнтовані СУБД.

Варто наголосити, що терміни «інформація» і «дані» суттєво відрізняються. Дані – це окремі факти, що характеризують об'єкти, процеси та явища предметної області, а також їхні властивості. Інакше кажучи, дані – це інформація, зафіксована в певній формі, яка є придатною для подальшої

обробки, зберігання та передавання (зберігається в базах даних, обробляється прикладними програмами тощо).

1.3 Поняття інформаційної системи та класифікація інформаційних систем

Інформаційна система («information system» (далі – ІС)) – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. Згідно з ДСТУ2392-94 Інформаційна система під ІС розуміють комунікаційну систему, що забезпечує збирання, пошук, оброблення та пересилання інформації.

Інформаційні системи можуть відрізнятись за типами об'єктів управління, різновидом і обсягом завдань тощо. Відповідно до цього, ІС варто класифікувати за певними ознаками.

1.3.1 Класифікація за рівнем або сферою діяльності

За рівнем (масштабом діяльності) розрізняють такі ІС: державні, регіональні, галузеві, системи об'єднань (підприємств або установ), технологічних процесів.

Державні ІС призначені для складання поточних і перспективних планів розвитку країни, для обліку результатів діяльності окремих ланок народного господарства, для регулювання цими ланками тощо. За допомогою ІС цього рівня розроблюють державний бюджет країни, контролюють його виконання. Прикладами ІС систем державного рівня є система управління національним банком (АСУ банк), автоматизована система державної статистики (АСДС), автоматизована система планових розрахунків (АСПР). державна ІС фінансових розрахунків (АСФР) при Міністерстві фінансів України, система обробки науково-технічної інформації (АСО НТІ) тощо.

Регіональні ІС орієнтовані на управління певним адміністративно-територіальним регіоном. Це може бути ІС області, міста, району. Такі системи обробляють інформацію, яка потрібна для управління регіоном, також вони

формують звітність, надають оперативні дані місцевим державним та господарським органам.

Галузеві системи орієнтовані на управління конкретними підприємствами та організаціями. Ці ІС працюють у промисловості та в сільському господарстві, будівництві й на транспорті тощо. Метою більшості галузевих систем є розв'язання завдань інформаційного обслуговування органів управління галузевих міністерств та їхніх підрозділів. Галузеві ІС поділяють за сферами застосування (промислова, непромислова, наукова тощо).

Інформаційні системи управління підприємствами (далі – ІСУП) або виробничими об'єднаннями (далі – ІСУ ВО) спрямовані на регулярне вирішення завдань управління виробничо-господарською діяльністю підприємств із застосуванням сучасних засобів автоматизованої обробки даних, економіко-математичних, статистичних методів, інструментів моделювання тощо.

Інформаційні системи управління технологічними процесами (далі – ІСУ ТП) мають на меті управління станом технологічних процесів (це може бути робот, плавильна піч, верстат тощо). Такі системи вирізняються двома важливими ознаками:

- особливий об'єкт управління – для ІСУ ТП це, насамперед, різноманітне обладнання;
- специфічну форму передавання інформації – в ІСУ ТП головною формою передавання інформації є сигнал, тоді як в інших ІСУ інформація передається через документи.

1.3.2 Класифікація за рівнем автоматизації процесів управління

Враховуючи мету та завдання, покладені на ІС, розрізняють пошукові, довідкові, управлінські (керівні), інтелектуальні інформаційні системи та системи підтримки прийняття рішень [1–3].

Інформаційно-пошукові системи (далі – ПС) зорієнтовані на розв’язування задач пошуку інформації. Змістова обробка інформації у таких системах незначна або відсутня.

В інформаційно-довідкових системах (далі – ІДС) за результатами пошуку здійснюються певні обчислення (тобто здійснюється первинна обробка даних). Зазвичай така обробка нескладна.

Управлінські (інформаційно-керівні) системи відомі у вітчизняній літературі, як «автоматизовані системи організаційного управління». Вони становлять організаційно-технічні системи, що забезпечують пошук управлінських рішень на базі автоматизації інформаційних процесів [1–3]. Ці системи мають на меті автоматизоване вирішення низки управлінських завдань.

До інформаційних систем нового покоління належать системи підтримки прийняття рішень (далі – СППР) та інформаційні системи, побудовані на основі штучного інтелекту (інтелектуальні ІС).

СППР – це інтерактивна комп’ютерна система, яка має на меті підтримку прийняття управлінських рішень за умов існування погано структурованих або взагалі неструктурованих проблем. Системи такого класу здатні суттєво підвищити ефективність праці в управлінні різними секторами економіки. Відповідно, зацікавленість цими системами постійно зростає, а розробка та реалізація СППР де-факто стала окремою галуззю інфо-бізнесу, який останнім часом показує чи не найбільшу динаміку розвитку.

Інтелектуальні ІС відрізняються тим, що в процесі своєї роботи імітують вирішення людиною складних, нетривіальних, навіть творчих завдань. Створенню таких систем сприяло розроблення в теорії штучного інтелекту т. з. логіко-лінгвістичних моделей. Ці моделі дають змогу формалізувати певні змістові відомості про об’єкти та процеси управління й надалі використовувати їх разом із «класичними» математичними моделями. Логіко-лінгвістичні моделі (семантичні мережі, фрейми, нейросистеми тощо) іноді об’єднують під

загальною назвою «програмно-апаратні засоби в системах штучного інтелекту».

Розрізняють три різновиди інтелектуальних ІС:

– інтелектуальні інформаційно-пошукові системи (типу «запитання – відповідь»); такі системи забезпечують взаємодію кінцевих користувачів (як правило, непрограмістів) із базами даних та знань у відповідній галузі. Спілкування із системою здійснюється у режимі діалогу з використанням спеціальних професійних мов користувачів, що близькі до природних;

– розрахунково-логічні інтелектуальні системи, які дають змогу кінцевим користувачам та спеціалістам із прикладної математики розв'язувати в діалоговому режимі певні завдання, пов'язані з використанням складних методів обробки даних із залученням відповідних прикладних програм;

– експертні системи (далі – ЕС), які дають змогу провадити ефективну комп'ютеризацію сфер, де знання можуть бути представлені в описовій, експертній формі, а використання математичних моделей ускладнене або взагалі неможливе.

Значною перевагою ЕС є їх здатність виявляти, накопичувати та узагальнювати відомості з різних галузей народного господарства (предметних областей).

1.3.3 Класифікація за ступенем централізації та інтеграції обробки даних

За ступенем централізації обробки даних розрізняють централізовані ІС, децентралізовані ІС, інформаційні системи колективного використання [1–4].

До централізованих ІС належать такі, де накопичення й обробка інформації проводиться в єдиному центрі. У цьому разі доступ до самої ІС здійснюється з одного або багатьох терміналів.

Децентралізовані ІС є більш автономними. Кожна децентралізована ІС певного рівня обслуговує певну верству користувачів. Прикладом децентралізованої ІС може слугувати система обробки статистичних даних. Для

неї характерно те, що ІС районного рівня обслуговує певний район, ІС обласного рівня обслуговує певну область тощо.

Інформаційні системи колективного користування відрізняються тим, що доступ до них мають багато користувачів.

Залежно від ступеня інтеграції функцій розрізняють багаторівневі ІС з інтеграцією за рівнями управління (підприємство – об'єднання, об'єднання – галузь тощо) і багаторівневі ІС з інтеграцією за рівнями планування.

Однорівневі інформаційні системи обслуговують переважно окремі підрозділи управління або виробництва. Однорівневою може бути система планово-фінансового відділу, автоматизована системи диспетчера виробництва, працівника складу тощо.

Багаторівневим інформаційним системам притаманні інтеграція функцій за різними аспектами управління. Зокрема, до багаторівневих здебільшого належать системи автоматизації бухгалтерського обліку, комплексного управління виробництвом, системи ресурсного забезпечення, загальнодержавні системи тощо.

1.3.4 Класифікація за типом інформаційної системи

Згідно за цим показником розрізняють фактографічні, документальні й документально-фактографічні ІС.

У документальній ІС об'єктом зберігання є власне документи.

Фактографічна ІС зберігає дані про об'єкти або сутності, що становлять певний інтерес для конкретної проблемної сфери (інформація про співробітників, договори, вироби тощо). Відомості про ці сутності можуть міститися у великій кількості таблиць, звітів, різних вхідних та вихідних повідомленнях.

1.3.5 Класифікація за сферою застосування

За цим показником ІС поділяють на такі групи:

- СОР – системи обробки транзакцій (протоколів);

- СППР – системи підтримки прийняття рішень;
- ІДС – інформаційно-довідкові системи;
- ОІС – офісні інформаційні системи.

Залежно від оперативності обробки даних СОР ділять на пакетні ІС та оперативні ІС.

Варто зазначити, що в інформаційних системах організаційного управління переважає режим оперативної обробки транзакцій (*Online Transaction Processing* або *OLTP*). Він дозволяє забезпечити високу оперативність відображення актуального стану предметної області на будь-який момент часу. Щодо пакетної обробки, то на сьогодні вона мало розповсюджена серед сучасних інформаційних систем організаційного управління.

Системи типу *OLTP* характеризуються регулярним, іноді досить інтенсивним потоком транзакцій. Це можуть бути замовлення, платежі, запити тощо, тому принциповими вимогами до таких систем є:

- висока продуктивність обробки транзакцій;
- висока надійність передачі інформації за умов віддаленого доступу до БД через систему телекомунікацій.

Системи підтримки прийняття рішень (іноді їх позначають *DSS*) відрізняються тим, що отримують на вході та обробляють досить складні запити. За результатами цих запитів СППР відбирають та аналізують дані у різних розрізах: за часом, географією, спеціальними характеристиками процесів тощо. Складність та розмаїття запитів потребує використання у складі СППР розвинутих СУБД (як правило, реляційного типу)

На відміну від СППР, інформаційно-довідкові системи здебільшого зорієнтовані на здатність працювати з інформацією довільної структури. Саме тому більшість таких систем базується на використанні гіпертекстових документів та мультимедіа. Найбільшого розвитку такі інформаційні системи набули в мережі Інтернет.

Клас ОІС здійснює переважно збереження, відтворення та обробку документів, а також переведення паперових документів в електронний вигляд. Серед функціональних можливостей цих систем переважають завдання автоматизації діловодства та управління документообігом.

1.3.6 Класифікація за способом організації

За способом організації інформаційні системи розподіляються на такі класи:

- на базі архітектури «файл-сервер»;
- на базі архітектури «клієнт-сервер»;
- на базі багаторівневої архітектури;
- на базі Інтернету або інтранет-технологій.

Зрозуміти обмеження різних архітектур допоможе аналіз функціональних компонентів будь-якої інформаційної системи та особливостей побудови її інформаційних додатків (табл. 1.2).

Таблиця 1.2 – Типові функціональні компоненти інформаційної системи

Позначення	Найменування	Характеристика
1	2	3
PS	<i>Presentation Services</i> (засоби представлення)	Обслуговує введення даних і запитів від користувача й відображає те, що повідомляє йому компонент логіки подання <i>PL</i> з використанням відповідної програмної підтримки
PL	<i>Presentation Logic</i> (логіка представлення)	Керує взаємодією між користувачем і системою. Обробляє дії користувача під час вибору команди в меню, при виборі пункту зі списку тощо
BL	<i>Business Logic</i> (прикладна логіка)	Реалізує набір правил для прийняття рішень, обчислень і операцій, які повинен виконати програмний додаток
DL	<i>Data Logic</i> (логіка управління даними)	Оперує з базою даних через мову <i>SQL</i> , які потрібно розглянути для реалізації прикладної логіки управління даними

Продовження таблиці 1.2

1	2	3
DS	<i>Data Services</i> (операції з базою даних)	Підтримує дії СУБД, що реалізують логіку управління даними, наприклад: маніпулювання даними, визначення даних, фіксація або повернення транзакцій тощо
FS	<i>File Services</i> (файлові операції)	Реалізує дискові операції читання і запису даних для СУБД та інших компонентів. Зазвичай є функціями операційної системи (ОС)

Архітектура «файл-сервер».

В архітектурі «файл-сервер» [2–4] розподіл компонентів діалогу PS і PL на мережному рівні відсутній, що певною мірою полегшує побудову графічного інтерфейсу. Комп'ютер використовується переважно для функцій відображення. Файл-сервер тільки отримує дані, внаслідок чого додаткові користувачі та їх додатки збільшують навантаження на головний сервер не суттєво.

Об'єктами розробки в файл-серверному додатку є компоненти, що визначають логіку діалогу *PL*, логіку обробки *BL* і управління даними *DL*. Додаток реалізується або у вигляді закінченого модуля, або у вигляді спеціального коду для подальшої інтерпретації.

Головний недолік файл-серверної архітектури: під час виконання запитів до бази клієнта можуть передаватися великі обсяги даних. Це суттєво перевантажує мережу, збільшує час реагування та знижує ефективність роботи системи загалом. Цей недолік значно обмежує використання архітектур типу «файл-сервер», особливо за умов організації віддаленого доступу до баз даних через повільні канали зв'язку.

Одним із варіантів усунення такого недоліку є віддалене управління файл-серверним додатком у мережі. Для цього у локальній мережі встановлюють сервер додатків, поєднаний з телекомунікаційним сервером (сервером доступу). Сервер додатків реалізує основний функціонал, а діалог

між віддаленими клієнтами здійснює сервер комунікацій. Однак у такому разі додатки не мають бути надто складними, оскільки це призведе до перевантаження сервера і для вирішення цієї проблеми знадобиться дуже потужна платформа сервера додатків.

Архітектура «клієнт-сервер».

В архітектурі «клієнт-сервер» [2–4] діє розподіл компонентів програми та баз даних і розташування їх там, де вони будуть працювати найбільш ефективно.

Архітектура «клієнт-сервер» дозволяє вирішити проблеми, притаманні файл-серверним додаткам. Це досягається внаслідок розподілу компонентів програми та розташування їх там, де вони можуть працювати найбільш ефективно. Відмінність архітектури «клієнт-сервер» полягає у тому, що тут є визначені сервери баз даних, які обробляють запити на мові *SQL (Structured Query Language)*. Сам текст запиту формує додаток клієнта та пересилає його для подальшої обробки до сервера. Таким чином, сервер баз даних організує пошук, сортування та агрегування інформації. А клієнтська частина здійснює подальшу обробку результату та його візуалізацію у вигляді форм або звітів. Внаслідок такої організації досягається мінімальний трафік по мережі (передаються здебільшого текстові запити). Інформаційна база максимально відокремлена від клієнтських додатків, тому можливе незалежне адміністрування баз даних без коригування програм з боку клієнта.

Більшість конфігурацій типу «клієнт-сервер» використовують дворівневу модель, де клієнт звертається до послуг сервера. Передбачається, що діалогові компоненти *PS* і *PL* розміщуються з боку клієнта, що спрощує реалізацію графічного інтерфейсу. Компоненти управління даними *DS* і *FS* розміщують на сервері, діалог (*PS*, *PL*) і логіка (*BL*, *DL*) – за клієнтом. У дворівневій архітектурі «клієнт-сервер» використовується саме такий варіант, коли додаток працює за клієнтом, а СУБД – за сервером.

Схема «клієнт-сервер» висуває найменші вимоги до сервера, тому вона має найкращу масштабованість. Однак складні додатки, що активно

взаємодіють з БД, можуть завантажити як клієнт, так і мережа. Результати *SQL*-запиту повинні повертатися клієнту для обробки, оскільки там реалізується логіка прийняття рішень. Це ускладнює адміністрування додатків, що розташовані в різних клієнтських вузлах.

Для зменшення навантаження на мережу і спрощення адміністрування додатками компонент *BL* також можна розмістити на сервері. У такому разі вся логіка прийняття рішень оформляється у вигляді збережених процедур і виконується на сервері БД.

Збережена процедура – послідовність *SQL*-операторів для організації доступу до БД. Збережена процедура має унікальне ім'я, може отримувати параметри і виконується на сервері БД. Для підвищення швидкості їх виконання збережені процедури можуть бути відкомпільовані.

Використання збережених процедур поліпшує рівень захищеності додатків, цілісність БД та безпеку системи загалом (немає безпосереднього доступу до даних), гарантує актуальність операцій та обчислень, що здійснюються колективно. Крім того, збережені процедури легше супроводжувати.

Реалізація архітектури «клієнт-сервер» можлива також на базі багатотермінальної системи. У цьому разі на стороні багаторівневого сервера додатків виконуються програми користувачів. Клієнтські вузли працюють як термінали. Подібна схема притаманна, наприклад, системі *Unix*.

Дворівневі схеми архітектури «клієнт-сервер» можуть додати певних проблем при реалізації складних додатків з великою кількістю користувачів. Вирішити ці проблеми можна шляхом застосування багаторівневої архітектури.

Багаторівнева архітектура.

Ця конфігурація є розвитком архітектури «клієнт-сервер». За класичною формою вона має три рівні:

– нижній – додатки клієнтів для виконання функцій і логіки *PS* і *PL*, які мають програмний інтерфейс для виклику програм середнього рівня;

– середній – сервер додатків, де виконується прикладна логіка *BL*; саме з цього сервера з допомогою логіки обробки *DL* здійснюються операції з базою даних *DS*;

– верхній – віддалений спеціалізований сервер для обробки даних *DS* і файлових операцій *FS* (без використання збережених процедур).

Подібну концепцію обробки даних підтримують, зокрема, фірми *Oracle*, *Sun*, *Borland* та ін.

Трирівнева архітектура дає змогу краще збалансувати навантаження на різні вузли та на мережу загалом, поліпшує спеціалізацію інструментів для розробки додатків, усуває недоліки дворівневої моделі «клієнт-сервер».

Централізація логіки додатків спрощує адміністрування і супровід системи. Чітко поділяються платформи та інструменти для реалізації інтерфейсу і прикладної логіки, що дозволяє фахівцям реалізовувати їх із максимальною користю. Окрім того, зміни прикладної логіки та інтерфейсу не залежать одне від одного. Межі між компонентами *PL*, *BL* і *DL* розмиті, тому прикладна логіка може реалізовуватися на всіх трьох рівнях. Сервер додатків за допомогою монітора транзакцій забезпечує інтерфейс із клієнтами та іншими серверами, може керувати транзакціями та підтримувати цілісність бази даних. Засоби віддаленого виклику процедур найбільше відповідають ідеї розподілених обчислень, оскільки вони забезпечують виклик прикладної процедури з будь-якого вузла мережі, розташованої в іншому вузлі.

Досвід останніх років розвитку систем «клієнт-сервер» довів доцільність використання трьох рівнів архітектури. Продукти для такої трирівневої архітектури (так звані монітори транзакцій) є відносно новими. Ці інструменти здебільшого орієнтовані на середовище *Unix*, однак прикладні сервери можна будувати також і на базі *Microsoft Windows*.

На практиці в локальній мережі можуть використовуватися змішані архітектури (дво- та трирівневі), а також архітектури з більшою кількістю рівнів.

Таким чином, багаторівнева архітектура дає змогу підвищити ефективність роботи корпоративної інформаційної системи й оптимізувати розподіл її програмно-апаратних ресурсів. Однак поки що на практиці домінує архітектура «клієнт-сервер».

Інтернет та інтранет-технології.

Донедавна головний акцент щодо розвитку Інтернету та інтранет-технологій був спрямований на розроблення інструментальних програмних засобів, тоді як нагальні потреби у розвинутих засобах розроблення додатків, що працюють з базами даних, були забезпечені не належним чином. Компромісним рішенням для вирішення цього питання стало об'єднання Інтернету та інтранет-технологій з багаторівневою архітектурою. У цьому разі структура інформаційного додатку набуває такого вигляду: браузер – сервер додатків – сервер баз даних – сервер динамічних сторінок – вебсервер.

За допомогою інтеграції Інтернету та інтранет-технологій із архітектурою «клієнт-сервер» процес упровадження та супроводження ІС значно спрощується, до того ж зберігається висока ефективність та можливість спільного використання інформації.

1.3.7 Вимоги до інформаційних систем

Незалежно від класу, структури та галузі використання ІС має відповідати таким вимогам: гнучкість, надійність, ефективність, безпека.

Гнучкість означає здатність системи до адаптації та подальшого розвитку. Це передбачає можливість пристосування інформаційної системи до нових умов та особливих потреб конкретного підприємства. Забезпечити гнучкість можна тільки за умови, що на етапі розроблення ІС були задіяні загальновідомі, науково обґрунтовані засоби й методи документування, внаслідок чого зберігається можливість розглянути структуру системи і внести до неї відповідні зміни.

Надійність ІС означає її стабільну роботу за будь-яких умов без спотворення інформації та без втрати даних. Високої надійності можна досягти

різними способами: через створення резервних копій, за рахунок операцій протоколювання, підтримання високої якості каналів зв'язку та фізичних носіїв інформації, використання сучасних програмних та апаратних засобів, забезпечення достатнього рівня кваліфікації персоналу.

Система є **ефективною**, якщо вона здатна вирішувати поставлені завдання за мінімального терміну, використовуючи при цьому надані їй ресурси. За будь-яких обставин ефективність ІС оцінює замовник. При здійсненні такої оцінки він має співставити обсяг засобів, вкладених у розробку, та відповідність ІС його очікуванням. Тільки активна участь замовника на всіх стадіях проєктування ІС дає змогу досягти високої ефективності та уникнути дискомфорту від впровадження нової технології в реальне виробництво.

Активна співпраця із замовником на ранніх етапах проєктування дозволяє з'ясувати його потреби і, внаслідок цього, зменшити витрати та термін розроблення системи. Крім того, контакт із замовником під час розроблення ІС може підштовхнути його до модернізації апаратних засобів, застосування нових методів ведення бізнесу, внесення змін до організаційної структури підприємства тощо. Це, зі свого боку, відповідає потребам як замовника, так і проєктувальника, оскільки замовник підвищує ефективність свого підприємства, а проєктувальник отримує нові можливості щодо проєктування ІС. Ефективність системи також забезпечується оптимізацією даних і методів їхньої обробки, застосуванням оригінальних алгоритмів, ідей, методів проєктування тощо.

Варто пам'ятати, що працювати з системою можуть малодосвідчені щодо роботи з інформаційними технологіями фахівці, тому оптимізація інтерфейсу відіграє не останню роль у забезпеченні високої ефективності системи на етапі реальної експлуатації. Інтерфейс має бути зрозумілим, структура команд меню та графічних елементів – ретельно проаналізована та оптимізована.

Під **безпекою**, передусім, розуміють властивість ІС контролювати доступ до інформації з боку різних категорій користувачів, забезпечувати цілісність

даних та їхній захист від випадкового пошкодження. Захист інформації від несанкціанованого доступу забезпечується розмежуванням на рівні ресурсів системи, використанням спеціальних програмних засобів та компонентів операційної системи щодо захисту інформації. У великих організаціях доцільно створювати окремі підрозділи для вирішення завдань інформаційної безпеки. У невеликих організаціях можна призначати окремого співробітника, відповідального за таку ділянку роботи.

Також необхідно зважати на можливе порушення безпеки внаслідок недосконалості програмного коду, вірусних атак, похибок в алгоритмах обробки даних тощо. Для вирішення подібних проблем організують постійний моніторинг стану безпеки системи та її оточення (операційного, прикладного, апаратного, спеціальних засобів захисту тощо).

1.4 Питання для самоконтролю

1. Подайте визначення терміна «інформаційна система».
2. Що таке структуровані дані? Наведіть приклади.
3. Чим відрізняються дані від інформації?
4. Як класифікують інформаційні системи за сферою застосування?
5. Що таке інформаційно-керівні системи та чим вони різняться?
6. Поясніть, що таке СППР.
7. Перелічіть різновиди інтелектуальних інформаційних систем.
8. Перелічіть види інформаційних систем за ступенем централізації.
9. Що таке документальна система?
10. Перелічіть різновиди інформаційних систем за способом організації.
11. Назвіть функціональні компоненти інформаційних систем.
12. Назвіть різновиди архітектур інформаційних систем.
13. Архітектура «файл-сервер», її переваги та недоліки.

14. Архітектура «клієнт-сервер».

15. Перелічіть головні вимоги до інформаційних систем.

2 ПРИНЦИПИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Будь-яка організація – це сукупність взаємодіючих елементів (підрозділів), що мають свою, досить складну, структуру та розгалужену систему зв'язків. Ці зв'язки можна розділити на три категорії [1–3]:

- функціональні, коли кожен підрозділ виконує певні різновиди робіт у межах єдиного бізнес-процесу;
- інформаційні, коли підрозділи обмінюються інформацією (документами, звітами, базами даних, розпорядженнями тощо);
- зовнішні, коли окремі підрозділи взаємодіють із зовнішнім середовищем, до того ж як на інформаційному, так і функціональному рівнях.

У цьому контексті процес розробки й упровадження інформаційної системи можна розглядати в двох аспектах:

- за змістом дій розробників, коли процес розробки розглядають, як певний статичний об'єкт у термінах потоків робіт (виконавці, дії, послідовність дій тощо);
- за часом або за стадіями життєвого циклу (далі – ЖЦ), коли за основу беруть динамічну організацію всього процесу розробки в термінах циклів, стадій, ітерацій та етапів.

Саме другий підхід на сьогодні є домінуючим при розробці складних ІС. Отже, далі будемо орієнтуватись на процес розробки ІС за другим варіантом.

2.1 Основні фази проєктування інформаційної системи

Незалежно від складності кожний проєкт проходить певні етапи. Стосовно ІС таких етапів п'ять [1–3]:

- формування концепції;
- підготування технічного завдання;
- проєктування ІС;
- розроблення ІС;
- введення ІС в експлуатацію.

2.1.1 Концептуальна фаза (формування концепції)

Головним завданням робіт на цьому етапі є визначення мети проєкту та його концепції, що передбачає:

- формування цілей та ідеї проєкту;
- формування команди для реалізації проєкту;
- вивчення вимог замовника щодо мотивації учасників проєкту;
- збір даних та аналіз наявного стану у контексті тематики проєкту;
- визначення ключових вимог і обмежень, обсягів матеріальних, фінансових і трудових ресурсів для реалізації проєкту;
- здійснення порівняльної оцінки альтернатив;
- подання пропозицій стосовно проєкту, проведення їхньої експертизи та затвердження.

2.1.2 Підготування технічного завдання

Головне завдання цього етапу – уточнення технічної пропозиції за результатами переговорів із замовником щодо укладання контракту. Зміст етапу зазвичай такий:

- розроблення основного змісту та базової структури проєкту;
- розроблення та затвердження технічного завдання;
- планування й подальша декомпозиція¹ базової структури проєкту;
- складання кошторису та бюджету проєкту, визначення потреб щодо ресурсів;
- розроблення календарних планів та узагальнених графіків виконання робіт;
- підписання контракту на виконання проєкту;
- упровадження засобів комунікації учасників проєкту та засобів контролю за процесом виконання робіт.

¹ Декомпозиція – один із головних способів підвищення ефективності робіт. Підсистеми, на які розбивається проєкт, мають бути слабо пов'язані за даними й за функціями. Кожна підсистема розробляється окремо.

2.1.3 Проєктування

Мета проєктування – з’ясувати склад підсистем, що підпадають під горизонт проєкту, визначити взаємозв’язки між ними, вибрати найбільш ефективні способи виконання проєкту й використання ресурсів. Фаза проєктування передбачає такі роботи:

- виконання базових проєктних робіт;
- розроблення індивідуальних технічних завдань;
- концептуальне проєктування;
- складання технічних специфікацій та інструкцій;
- представлення проєкту, його експертиза та затвердження.

2.1.4 Розроблення

Головні завдання цієї фази – розробити механізми координації та оперативного контролю за роботами, що передбачені проєктом, реалізувати весь склад підсистем, забезпечити їх об’єднання, взаємодію та тестування. Основний зміст фази розробки полягає в такому:

- розроблення програмного забезпечення;
- підготування до впровадження системи;
- перевірка та регулювання головних показників проєкту.

2.1.5 Уведення системи в експлуатацію

На етапі введення системи в експлуатацію проводяться випробування окремих компонент та ІС загалом, дослідна експлуатація системи в реальних умовах, переговори щодо результатів виконання проєкту, можливих змін та нових контрактів.

Ключові роботи цього етапу:

- комплексні випробування;
- підготування кадрів для експлуатації системи;
- підготування робочої документації (далі – РД), здавання системи замовнику і введення її в експлуатацію;
- супроводження, підтримка, сервісне обслуговування;

- оцінювання результатів проєкту та підготування підсумкових звітів;
- вирішення конфліктних ситуацій і закриття робіт за проєктом;
- накопичення дослідних даних для подальших проєктів, аналіз досвіду, стану, визначення напрямів розвитку.

Варто зауважити, що пошук помилок на стадії проєктування забирає приблизно вдвічі більше часу, ніж на інших етапах, а їх виправлення коштує у п'ять разів дорожче. Саме тому розроблення проєкту на початкових стадіях варто виконувати дуже ретельно. Типові помилки, які роблять розробники на початкових стадіях проєкту:

- помилки у визначенні інтересів замовника;
- зосередження на другорядних інтересах;
- хибна інтерпретація початкового завдання;
- помилкове або недостатнє усвідомлення деталей;
- неповнота функціональних специфікацій;
- вади у визначенні необхідних ресурсів і термінів виконання робіт;
- недостатня перевірка узгодженості етапів, відсутність контролю з боку замовника.

2.2 Процеси, що відбуваються під час життєвого циклу інформаційної системи

Життєвий цикл – це безперервний процес, що починається з моменту прийняття рішення про створення ІС і закінчується після вилучення її з експлуатації [1–3]. Регламентує життєвий цикл інформаційних систем міжнародний стандарт ISO/IEC 12207. 2.

Стандарт ISO/IEC 12207 визначає структуру життєвого циклу, його процеси, дії, завдання, які мають бути виконані під час створення ІС. Згідно з цим стандартом структура життєвого циклу включає три групи процесів:

² Скорочення ISO походить від *International Organization of Standardization* (міжнародна організація із стандартизації), IEC – від *International Electrotechnical Commission* (міжнародна комісія з електротехніки).

- **головні** – придбання, постачання, розроблення, експлуатація, супроводження;
- **допоміжні** – забезпечують виконання головних процесів (документування, забезпечення якості, тестування, атестація, оцінка ефективності, аудит, ліцензування, сертифікація тощо);
- **організаційні** – управління проектами, створення інфраструктури проекту, навчання.

2.2.1 Головні процеси життєвого циклу

Життєвий цикл ІС включає три основні процеси: розроблення, експлуатація та супровід (супроводження).

Розроблення. Цей процес поєднує всі роботи щодо створення інформаційного та програмного забезпечення та його компонентів відповідно до вимог техзавдання. Розроблення інформаційного програмного забезпечення включає також такі питання:

- оформлення проектної та експлуатаційної документації (далі – ЕД);
- підготування матеріалів для тестування програмних продуктів;
- розроблення матеріалів для навчання персоналу.

Розроблення – один із найважливіших процесів життєвого циклу ІС. Зазвичай він передбачає стратегічне планування, аналіз, проектування та реалізацію програмних продуктів.

Експлуатація.

Умовно роботи на етапі експлуатації можна поділити на **головні** та **підготовчі**. До головних робіт належать:

- локалізація проблем та усунення причин їхнього виникнення;
- модифікація (або адаптація) програмного забезпечення;
- підготування пропозицій щодо вдосконалення системи;
- розвиток та подальша модернізація системи.

Підготовчі роботи зазвичай включають:

- конфігурацію бази даних і робочих місць користувачів;

- забезпечення користувачів експлуатаційною документацією;
- навчання (або підвищення кваліфікації) персоналу.

Супроводження.

Технічна підтримка – дуже важливий складник успішної експлуатації ІС, необхідна умова вирішення поставлених перед нею завдань. Варто зауважити, що помилки обслуговуючого персоналу можуть призводити до фінансових втрат, які можна порівняти з вартістю самої системи.

Технічна підтримка – досить складний та довготривалий процес, тому для його успішної реалізації дуже важливий організаційний складник, коли потрібно виконати такі дії:

- виокремити ключові складники системи, оптимізувати розподіл ресурсів для їхнього технічного обслуговування (далі – ТО);
- визначити завдання щодо ТО, розділити їх на внутрішні (що вирішуються власними силами) та зовнішні (вирішуються спеціалізованими сервісними організаціями); забезпечити чіткий розподіл функцій та відповідальності;
- провести аудит ресурсів, необхідних для організації ТО, розподілити обов'язки;
- підготувати план організації ТО, у якому конкретизувати етапи дій, терміни виконання, витрати за кожним етапом, відповідальність виконавців.

Якісне технічне обслуговування потребує залучення висококваліфікованих фахівців, які здатні не тільки вирішувати поточні завдання, а й швидко відновлювати дієвість системи в аварійних ситуаціях.

2.2.2 Допоміжні процеси життєвого циклу

Серед допоміжних провідне місце займає процес управління конфігурацією, який підтримує головні етапи життєвого циклу ІС, зокрема процеси розроблення та супроводження.

Під час розроблення складних ІС кожен компонент може розглядатися окремо, мати декілька варіантів або версій реалізації. У такому випадку

виникає проблема щодо обліку їхніх зв'язків і функцій, створення єдиної структури для забезпечення розвитку всієї системи. Цю проблему вирішує управління конфігурацією, що дозволяє системно контролювати внесення змін у різні компоненти ІС на всіх стадіях ЖЦ.

2.2.3 Організаційні процеси

Управління проектом тісно пов'язане з плануванням і організацією робіт, створенням колективу проєктувальників, з контролем за термінами та якістю реалізації всіх етапів розробки ІС. Вирішення цих питань покладається на технічне й організаційне забезпечення проєкту, яке передбачає:

- вибір методів та інструментальних засобів для реалізації проєкту;
- визначення методів щодо опису всіх проміжних станів розробки;
- розроблення методів і засобів для випробувань ПЗ;
- навчання та підвищення кваліфікації персоналу.

Забезпечення якості проєкту тісно пов'язане з проблемами **верифікації**³, перевірки й тестування компонентів ІС.

Перевірка – це процес, мета якого – визначення відповідності параметрів ІС вихідним вимогам. Перевірка певною мірою подібна до тестування, яке має на меті визначення розбіжностей між дійсними та очікуваними результатами проєкту та оцінкою відповідності характеристик ІС вихідним вимогам.

2.3 Структура життєвого циклу інформаційної системи

Повний життєвий цикл ІС передбачає стратегічне планування, аналіз, проєктування, реалізацію, упровадження та експлуатацію. Крім того, життєвий цикл можна розбити на декілька стадій. Один із варіантів такого розподілу запропонувала корпорація *Rational Software* – провідна фірма на ринку

³ Верифікація – процес визначення відповідності поточного стану розробки вимогам до цього етапу.

програмного забезпечення засобів щодо розроблення інформаційних систем (зокрема універсального CASE-засобу *Rational Rose*)⁴.

Згідно з методологією *Rational Software* ЖЦ ІС має чотири стадії:

- початок;
- уточнення;
- конструювання;
- передавання для експлуатації.

Межі кожної стадії визначаються певними моментами часу, коли необхідно приймати критичні рішення і досягати певних цілей.

2.3.1 Початкова стадія

На цій стадії визначають область застосування системи та граничні умови для майбутньої розробки. Для вирішення такого завдання необхідно ідентифікувати всі зовнішні об'єкти, з якими буде взаємодіяти система, визначити особливості цієї взаємодії. Крім того, на початковій стадії ідентифікують усі функціональні можливості системи й роблять опис найістотніших із них. Також початкова стадія передбачає:

- критерії успіху розробки;
- оцінювання ризику;
- оцінювання ресурсів, необхідних для виконання розробки;
- календарний план із зазначенням термінів завершення головних етапів.

⁴ Первісне значення терміна *CASE* (*Computer Aided Software/System Engineering*) обмежувалось тільки питаннями автоматизації розроблення програмного забезпечення. Однак потім значення цього терміна стало ширшим. Зараз під терміном «CASE-засобу» розуміють програмні засоби, що підтримують процеси створення і супроводження інформаційних систем, зокрема аналіз і формулювання вимог, проектування прикладного програмного забезпечення та баз даних, генерацію коду, тестування, документування, забезпечення якості, конфігураційне управління та управління проектом, а також інші процеси.

2.3.2 Стадія уточнення

На стадії уточнення проводять аналіз прикладної області та розробляють архітектуру майбутньої ІС. На завершальному етапі цієї стадії проводять аналіз архітектурних рішень і способів усунення головних чинників ризику в проєкті.

2.3.3 Стадія конструювання

На стадії конструювання розробляють закінчений продукт, готовий до передачі в експлуатацію. Після закінчення цієї стадії оцінюють працездатність програмного забезпечення.

2.3.4 Стадія передавання до експлуатації

На цій стадії готове програмне забезпечення передають користувачу (або користувачам). Варто зауважити, що в процесі експлуатації системи можуть з'явитися різні проблеми, що потребують додаткових робіт щодо внесення коригувань до готового продукту. Отже, одне із завдань стадії експлуатації – виявлення необхідних коригувань, планування і проведення робіт щодо їх реалізації, а також внесення відповідних змін. Наприкінці стадії передавання для експлуатації визначають ступінь досягнення цілей розробки.

2.4 Моделі життєвого циклу інформаційної системи

Модель життєвого циклу – це певна структура, що визначає послідовність та взаємозв'язки між процесами, що передбачені в межах ЖЦ інформаційної системи.

Ключові положення і базові визначення моделі ЖЦ викладені в стандарті ISO/IEC 12207. Потрібно зазначити, що вказаний стандарт описує тільки структури процесів. У ньому немає деталізації методів і дій для вирішення завдань, що належать до процесів ЖЦ ІС. І це зрозуміло: регламенти ISO/IEC 12207 є загальними для будь-яких моделей і технологій розробки. Водночас сама модель ЖЦ залежить від специфіки ІС та умов, де вона створюється і буде працювати.

На сьогодні поширення набули дві головні моделі ЖЦ:

- каскадна (так звана «модель водоспаду» або «waterfall»);
- спіральна.

2.4.1 Каскадна модель життєвого циклу інформаційної системи

Згідно з каскадною моделлю всі роботи виконуються. При цьому вся розробка розбивається на етапи, перехід до наступного етапу відбувається після повного завершення всіх робіт попереднього етапу.

Кожен етап завершується оформленням повного комплексу документації. Склад і зміст цієї документації передбачає, що реалізація проєкту може бути продовжена іншою командою розробників.

Головні етапи розроблення згідно з каскадною моделлю.

Враховуючи досвід реальних розробок, у каскадній моделі прийнято виділяти декілька сталих етапів, що майже не залежать від предметної області (рис. 2.1), а саме:

- аналіз вимог замовника;
- проєктування;
- розроблення;
- тестування і дослідна експлуатація;
- здавання готового продукту.

Результатом першого етапу є ТЗ (завдання на розробку). Воно має бути узгоджене з усіма сторонами.

Результат другого етапу – комплект проєктної документації (далі – ПД), що містить необхідні дані для реалізації проєкту.

Результат третього етапу – готовий програмний продукт.

На четвертому етапі визначають приховані недоліки, які виявились за умов експлуатації, а також вносять відповідні коригування до програмного забезпечення.

Головне завдання п'ятого етапу – переконати замовника у тому, що всі вимоги виконані повною мірою.

На практиці ЖЦ реальної системи може істотно відрізнятись, бути складнішим і довшим. Він може містити довільну кількість циклів уточнення, змін і доповнень вже реалізованих проєктних рішень. до того ж саме у таких циклах відбувається розвиток ІС й модернізація її компонентів.

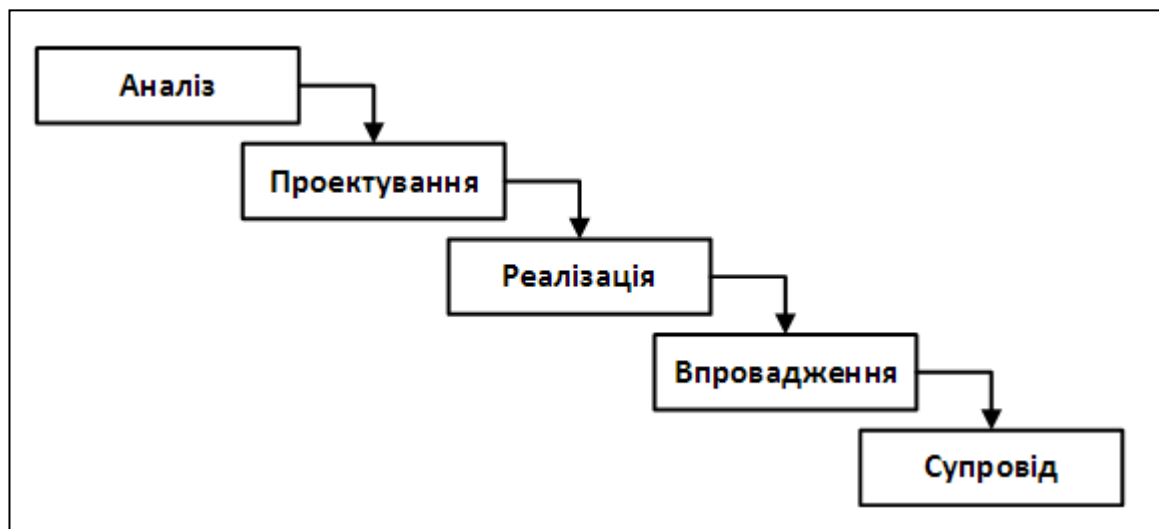


Рисунок 2.1 – Структурна схема каскадної моделі розробки ІС

Переваги каскадної моделі.

Серед переваг каскадної моделі передусім можна назвати такі:

- на кожному етапі формується повний та узгоджений набір ПД. На завершальних етапах також розробляється документація, що охоплює всі передбачені стандартами різновиди забезпечення ІС (організаційне, методичне, інформаційне, програмне, апаратне);
- чітка послідовність етапів дозволяє планувати терміни виконання робіт та відповідні витрати.

Каскадний підхід добре проявив себе під час розроблення певного класу ІС. Насамперед це системи, де з самого початку можна окреслити всі вимоги, що надає розробникам свободу щодо шляхів реалізації.

Недоліки каскадної моделі.

Кількість недоліків каскадної моделі досить значна, а саме:

- повільне отримання результатів;

- помилки на будь-якому етапі впливають на подальшу роботу;
- у межах каскадної моделі складно організувати паралельне ведення робіт;
- спостерігається певна інформаційна перенасиченість на всіх етапах проекту;
- управління проектом ускладнене;
- рівень ризику та ненадійність інвестицій значні.

Затримка в отриманні результатів – головний недолік каскадної моделі, що є наслідком послідовного підходу, адже в цьому випадку узгодження результатів проводиться тільки після завершення чергового етапу робіт.

Крім того, узгодження результатів зазвичай проводиться вибірково, після завершення кожного етапу, вимоги до ІС «заморожені» у вигляді технічного завдання на весь час її створення, тому користувачі можуть надати свої пропозиції та зауваження тільки після завершення робіт над системою. У разі неточного формулювання вимог або їхньої зміни за період створення ПО користувачі отримують систему, що не задовольняє їхні потреби.

До того ж задіяні на початок розробки моделі об'єктів можуть застаріти (унаслідок змінювання законодавства, в організаційній структурі об'єкта тощо). Це стосується всіх складників проекту: функціонального, інформаційних моделей, інтерфейсу користувача й документації.

Повернення до попередніх стадій. Цей недолік каскадної моделі є наслідком попереднього. Повернення може стати причиною порушення графіка робіт та ускладнити відносини між розробниками ІС, що працюють над різними етапами тощо. Найгіршим є той факт, що недоліки попереднього етапу можуть виявитися набагато пізніше. Унаслідок цього роботу треба припинити і повертатися з поточного до попереднього етапу, тому в реальному житті каскадна схема розробки виглядає так, як це показано на рисунку 2.2.

Складність паралельного виконання робіт. За каскадної моделі робота над проектом будується, як низка послідовних кроків. Навіть тоді, коли

розробку окремих частин (підсистем) можна виконувати паралельно, зробити це у рамках каскадної схеми складно. Це пов'язано з необхідністю постійно узгоджувати різні частини проєкту. Парадокс полягає в тому, що чим більш незалежними є частини, тим ретельніше має виконуватися синхронізація і, як наслідок, – тим більше залежать одна від одної групи розробників.

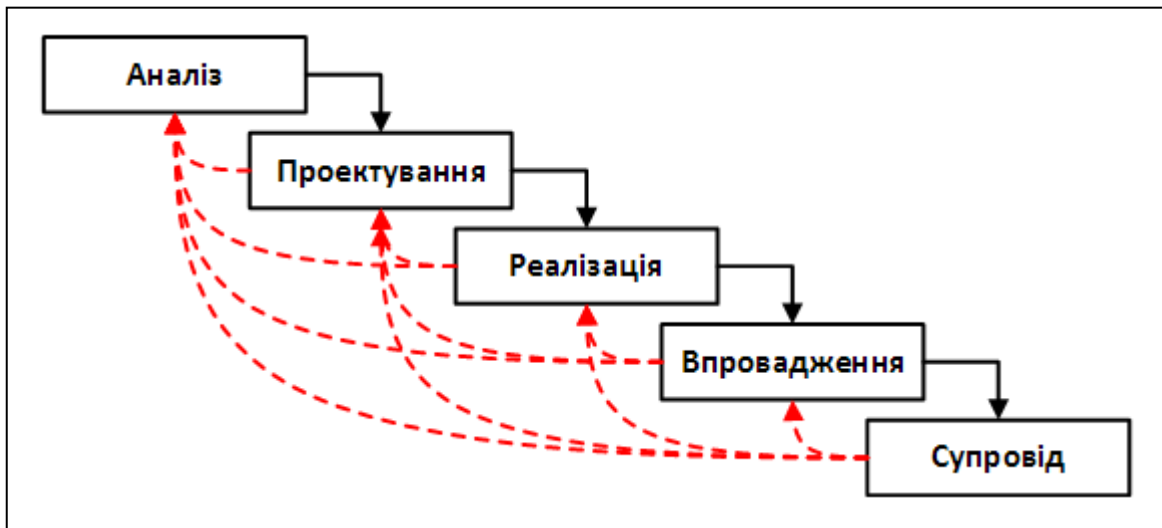


Рисунок 2.2 – Реальний процес розробки за каскадною схемою

Інформаційна перенасиченість. Цей недолік є наслідком суттєвої залежності між різними групами розробників. Проблема полягає в тому, що при внесенні змін до будь-якої частини проєкту потрібно повідомити про них усім розробникам, що працюють із цією частиною. Як наслідок, швидко зростають витрати на документування змін, опрацювання цих змін, погіршується оперативність узгоджень тощо і, загалом, збільшується обсяг інформації, що є результатом особливостей схеми управління проєктом, тобто його, в певному розумінні, «паразитної» природи.

Питання перенасиченості різко загострюється за умови ротації груп розробників, адже нові спеціалісти, крім вивчення нового матеріалу, повинні опанувати багато попередньої інформації, що пов'язана з історією змін, коригувань та узгоджень. Цей факт вкрай негативно впливає на ефективність проектування.

Складність управління проектом під час використання каскадної схеми здебільшого обумовлена чіткою послідовністю стадій розроблення та наявністю складних взаємозв'язків між різними частинами проекту.

Послідовність розроблення проекту призводить до того, що одні групи розробників повинні очікувати результати роботи інших команд. Отже, потрібне адміністративне втручання для узгодження термінів роботи та складу переданої документації.

Значний рівень ризику. Чим складнішим є проект, тим довшою буде тривалість кожного з його етапів, тим складнішими будуть взаємозв'язки між частинами проекту. Унаслідок специфіки каскадної моделі така ситуація може призвести до великої кількості повернень до попередніх етапів після виявлення змін та їх погодження.

Фактично це означає, що існує ймовірність значно збільшити терміни виконання проекту, а таке збільшення (з фінансової точки зору) означає високий рівень ризику інвестицій.

Нарешті, занадто велика кількість змін (особливо в предметній області або у вимогах замовника) можуть взагалі призупинити проект, не довівши його до остаточної реалізації.

Отже, можна стверджувати, що складні проекти, які розробляються за каскадною схемою, мають підвищений рівень ризику.

Область застосування. Каскадний підхід добре зарекомендував себе при розробці:

- 1) однорідних ІС, де кожен додаток становить єдине ціле;
- 2) ІС, для яких на самому початку розроблення можна досить чітко й повно сформулювати всі вимоги, щоб надати розробникам свободу для їх якнайшвидшої реалізації з технічного погляду. До цієї категорії потрапляють ІС зі складними обчисленнями, системи, що працюють у реальному часі і подібні.

На базі каскадної моделі життєвого циклу побудовано один із найвідоміших засобів автоматизації процесів розроблення складних інформаційних систем від компанії *Oracle*, що називається *Oracle Designer*.

Методика (в літературі вона має назву CRM), що використовується під час роботи цього комплексу, базується на таких положеннях:

1) проектування має структурне походження, весь процес розроблення системи подається у вигляді послідовності чітко визначених етапів;

2) підтримка здійснюється на всіх етапах життєвого циклу системи, починаючи від загальних пропозицій і закінчуючи супроводом готового продукту;

3) перевага надається архітектурі «клієнт-сервер», зокрема складним структурам розподілених баз даних;

4) під час розроблення всі специфікації проєкту зберігаються в спеціальній базі даних (репозиторії), яка працює під управлінням СУБД *Oracle*. До репозиторію можуть підмикатися багато користувачів (розробників), унаслідок чого їхні дії стають узгодженими;

5) послідовний перехід від одного етапу до іншого автоматизований унаслідок використання спеціальних утиліт. За допомогою них за специфікаціями на концептуальній стадії можна отримати початковий варіант специфікації рівня проектування. Надалі генерація доповнень значно спрощується;

б) усі етапи проектування та розробки автоматизовані; у будь-який момент може бути згенерований довільний обсяг звітів, які забезпечують документування поточної версії системи на всіх етапах її розроблення.

Oracle пропонує свій варіант структури життєвого циклу інформаційної системи, а методологія та програмні засоби щодо її реалізації зорієнтовані переважно на продукти цієї компанії. Незважаючи на певні недоліки та обмеження, технологія *CDM* залишається однією з провідних під час розроблення складних інформаційних систем.

2.4.2 Спіральна модель життєвого циклу

На відміну від каскадної спіральна модель ЖЦ передбачає ітераційний процес розробки ІС. До того ж у рамках цієї моделі зростає роль початкових

етапів ЖЦ (аналізу та проєктування) – саме на цих етапах перевіряється та обґрунтовується реалізація технічних рішень через створення прототипів.

Поняття ітерації.

Ітерація – це основний елемент концепції спіральної моделі. Кожна ітерація є завершеним циклом розробки, що призводить до випуску діючої версії виробу (або певної його частини). Надалі від ітерації до ітерації цей продукт вдосконалюється й наприкінці перетворюється у завершену систему (рис. 2.3).

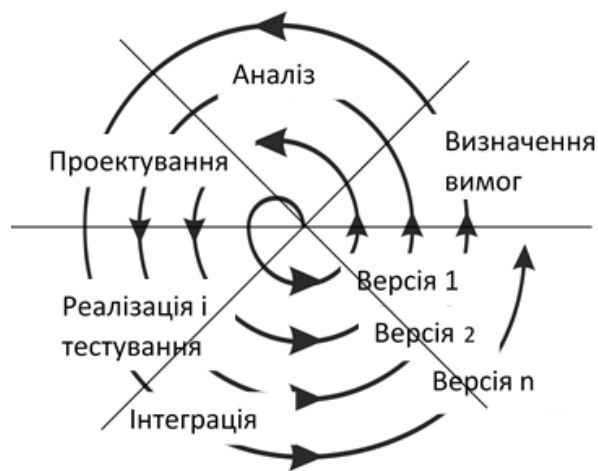


Рисунок 2.3 – Спіральна модель життєвого циклу інформаційної системи

Таким чином, кожне коло спіралі відповідає за створення частки або версії програмного продукту, на цьому ж колі уточнюються мета й характеристики проєкту, вимоги до якості, плануються роботи для наступного витка спіралі. На кожній ітерації послідовно конкретизуються деталі проєкту, унаслідок чого вибирається обґрунтований варіант. Потім цей варіант доводиться до остаточної реалізації. Використання спіральної моделі дає змогу здійснювати перехід до наступного етапу реалізації проєкту, не очікуючи повного завершення поточного етапу робіт – їх можна буде виконати у наступній ітерації.

Головне завдання кожної ітерації – якнайшвидше отримати діючий продукт, який можна показати користувачам системи. Ця обставина робить набагато простішим процес внесення уточнень і доповнень до проєкту.

Переваги спіральної моделі.

Спіральний підхід до розроблення програмного **забезпечення** дозволяє подолати більшість недоліків каскадної моделі. Крім того, він забезпечує безліч додаткових можливостей, роблячи процес розробки гнучкішими. Переваги ітераційного підходу такі:

1. Ітераційна розробка полегшує коригування проєкту у разі зміни вимог замовника.

2. У межах спіральної моделі окремі елементи ІС інтегруються до кінцевого продукту поступово, тобто фактично інтеграція триває безперервно. Оскільки інтеграція починається з меншої кількості елементів, виникає набагато менше проблем під час її проведення.

3. Під час реалізації проєкту за спіральною схемою зменшується рівень ризиків. Ця перевага є наслідком попередньої, оскільки ризики можна виявити вже на етапі інтеграції. Через це рівень ризиків є максимальним на початку розроблення проєкту. З часом, у міру просування розробки, очікуваний рівень ризиків зменшується.

4. Ітераційність організації робіт забезпечує гнучкість управління проєктом. Це дає змогу вносити тактичні зміни в процесі виконання робіт на будь-якому етапі.

5. Ітераційний підхід краще пристосований до повторного використання компонентів. Це обумовлено тим, що легше знайти загальні частини проєкту, коли вони вже частково розроблені, ніж намагатися відшукати їх на початковій стадії проєкту. Практика свідчить, що аналіз проєкту вже після декількох ітерацій дозволяє виявити компоненти багаторазового використання, які на наступних ітераціях будуть удосконалюватися.

6. Спіральна модель дає змогу отримати більш надійну та сталу систему. Це є наслідком того, що з розвитком системи помилки та слабкі місця виявляються і коригуються на кожній ітерації.

7. Ітераційний підхід дає змогу вдосконалити процес розробки. Аналіз, проведений наприкінці кожної ітерації, дозволяє оцінити, що має бути змінено в організації розробки та як її поліпшити на наступній ітерації.

Недоліки спіральної моделі.

Головна проблема спірального підходу – визначення моментів переходу між етапами. Для її вирішення необхідно вводити тимчасові обмеження на кожен із етапів ЖЦ. В іншому разі процес розробки може перетворитися на нескінченне вдосконалення зробленого, тому завершення ітерації має здійснюватися тільки відповідно до плану, навіть якщо не весь запланований обсяг робіт вже завершено. Щодо самого плану, то він складається на підставі статистичних даних попередніх проєктів та особистого досвіду розробників.

На базі спіральної моделі була створена потужна методологія швидкого розроблення додатків, що називається *RAD* (з англ. *Rapid Application Development*). Вона зорієнтована переважно на роботу невеликої команди програмістів, які мають досвід аналізу, проєктування, генерації коду й тестування ПЗ із використанням *CASE*-засобів.

Під час проєктування за методологією *RAD* активно використовуються об'єктно-орієнтовані методи опису предметної області. Методологія добре підходить для проєктів із коротким та добре опрацьованим графіком робіт. Серед відмінностей *RAD* можна назвати такі:

- розроблення програм ітераціями;
- допустимість часткового завершення робіт на кожному з етапів ЖЦ;
- обов'язкове залучення користувачів до процесу розроблення ІС;
- обов'язкове використання *CASE*-засобів для забезпечення цілісності проєкту;
- використання засобів управління конфігурацією, що полегшують внесення змін до проєкту й супровід готової системи;
- використання генераторів коду;

- прототипування, що дозволяє краще з'ясувати й задовольнити потреби кінцевого користувача;
- тестування і коригування проєкту одночасно з його розробленням;
- проведення розроблення невеликої, але добре керованої команди професіоналів (зазвичай від 2 до 10 осіб);
- професійне керівництво розробкою системи, чітке планування і контроль за виконанням робіт.

Життєвий цикл ПЗ за методологією *RAD* має чотири фази:

- 1) аналіз і планування вимог;
- 2) проєктування;
- 3) побудова;
- 4) упровадження.

2.4.3 Ітераційний підхід до моделі життєвого циклу

Розвиток каскадної та спіральної моделей привів до їхнього зближення. Результатом такого зближення стала поява сучасного ітераційного підходу, який фактично становить раціональне поєднання цих двох моделей.

Різні варіанти ітераційного підходу реалізовані в більшості сучасних методів: *Rational Unified Process (RUP)*, *Framework Microsoft Solutions (MSF)*, *XR* тощо:

1. Метод *RUP* (запропонований компанією *Rational Software* у 2003 р.) використовує ітеративну модель розроблення з чотирьох фаз (початок, дослідження, побудова, упровадження), розподілених на ітерації. Кожна ітерація завершується отриманням проміжної, але діючої версії кінцевого продукту. *RUP* спирається на інтегрований комплекс інструментальних засобів *Rational Suite*, до складу якого, крім самої технології *RUP* як продукту, входять такі компоненти:

- *Rational Rose* – засіб візуального моделювання (аналізу і проєктування), використовує мову *UML*;

– *Rational XDE* – засіб аналізу і проєктування, інтегрується з платформами *MS Visual Studio .NET*, *IBM WebSphere Studio Application Developer* та ін.

2. *MSF (Microsoft Framework, 1993 р.)* подібна до *RUP*, так само включає чотири фази: аналіз, проєктування, розробка, стабілізація, є ітераційною і передбачає використання об'єктно-орієнтованого моделювання. *MSF*, на відміну від, *RUP* більшою мірою зорієнтована на розробку бізнес-додатків.

3. Методологія *XP (Extreme Programming* або екстремальне програмування, 1996 р.). Розробка виглядає як ітеративний процес, де фази розбиваються на малі кроки. В основі методології – командна робота й ефективна комунікація між замовником та виконавцем.

2.5 Питання для самоконтролю

1. Назвіть головні різновиди зв'язків між підрозділами підприємства.
2. Перелічіть фази проєктування інформаційної системи.
3. Назвіть головне завдання та надайте перелік робіт концептуальної фази проєктування інформаційної системи.
4. У чому полягає зміст етапу «підготовка технічної пропозиції»?
5. Мета й завдання етапу «Проєктування».
6. Мета, завдання за головні складники етапу «Розроблення».
7. Мета, завдання, головні складники та відмінності етапу «Вступ в експлуатацію».
8. Типові помилки проєктування інформаційних систем.
9. Подайте визначення терміна «життєвий цикл інформаційної системи».
10. Перелічіть головні процеси життєвого циклу інформаційної системи.
11. Перелічіть головні процеси етапу «Розроблення».
12. Перелічіть головні процеси етапу «Експлуатація».

13. Перелічіть головні процеси етапу «Супровід».
14. Що таке «допоміжні процеси життєвого циклу»? Перелічіть їх та надайте стислу характеристику.
15. Що передбачає процес «Організація»?
16. Охарактеризуйте структуру життєвого циклу за концепцією *Rational Software*.
17. Склад робіт початкового етапу життєвого циклу.
18. Склад робіт етапу «Уточнення життєвого циклу».
19. Склад робіт етапу «Конструювання життєвого циклу».
20. Склад робіт етапу «Передача в експлуатацію життєвого циклу».
21. Різновиди моделей життєвого циклу інформаційної системи.
22. Каскадна модель життєвого циклу. Характеристика каскадної моделі, її особливості, переваги та недоліки.
23. Спіральна модель життєвого циклу. Характеристика спіральної моделі. Її особливості, переваги та недоліки.
24. Ієрархічна модель життєвого циклу. Характеристика ієрархічної моделі, її особливості, переваги, недоліки та сфера використання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Николайчук Я. М. Проектування спеціалізованих комп'ютерних систем : навч. посіб. / Я. М. Николайчук, Н. Я. Возна, І. Р. Пітух. – Тернопіль : ТзОВ «Терно-граф», 2010. – 392 с.
2. Шаховська Н. Б. Проектування інформаційних систем : навч. посіб. / Н. Б. Шаховська, В. В. Литвин. – Львів : Магнолія-2006, 2011. – 380 с.
3. Разработка программных проектов на основе Rational Unified Process (RUP) / Г. Поллис, Л. Огастин, К. Лоу, Дж. Мадхар. – Бином-Пресс, 2011. – 256 с.
4. Грекул В. И. Проектирование информационных систем : учеб. пособ. / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – М. : Интернет-Университет информационных технологий (ИНТУИТ.РУ) : БИНОМ. Лаборатория знаний, 2010. – 299 с.
5. Инюшкина О. Г. Проектирование информационных систем (на примере методов структурного системного анализа) : учеб. пособ. / О. Г. Инюшкина. – Екатеринбург : «Форт-Диалог Исеть», 2014. – 240 с.

Електронне навчальне видання

КАРПЕНКО Микола Юрійович

**ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ В УПРАВЛІННІ
ОРГАНІЗАЦІЄЮ**

КОНСПЕКТ ЛЕКЦІЙ
Частина 1 (ЗМ 1)

*(для здобувачів другого (магістерського) рівня вищої освіти денної і заочної
форм навчання зі спеціальності 073 – Менеджмент
і 071 – Облік і оподаткування)*

Відповідальний за випуск *О. Б. Костенко*
Редактор *О. А. Норик*
Комп'ютерне верстання *М. Ю. Карпенко*

План 2024, поз. 111Л

Підп. до друку 01.03.2024. Формат 60 × 84/16.
Ум. друк. арк. 2,7.

Видавець і виготовлювач:
Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Маршала Бажанова, 17, Харків, 61002.
Електронна адреса: office@kname.edu.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 5328 від 11.04.2017.