

Міністерство освіти та науки України
Чернівецький національний університет
імені Юрія Федьковича

Інформаційні системи та технології у системному аналізі

**Конспект лекцій та завдання
до лабораторних робіт**

Чернівці
Чернівецький національний університет
2024

УДК 004.7:517](076)

I-741

Рекомендовано до друку Вченою радою факультету
математики та інформатики Чернівецького національного університету
імені Юрія Федьковича
(протокол № 4 від « 23 » жовтня 2024 року)

I-741 Піддубна Л.А. Інформаційні системи та технології у систем-
ному аналізі. Конспект лекцій та завдання до лабораторних
робіт/ Укл: Л.А. Піддубна, – Чернівці : Чернівецький нац. ун-т,
2024. – 40 с.

Видання містить короткий конспект лекцій та завдання до лабора-
торних робіт, які охоплюють основні принципи виконання ІТ проектів.
Для студентів спеціальностей «Системний аналіз».

УДК 004.7:517](076)

© Піддубна Л.А. 2024

© Чернівецький національний університет, 2024

Зміст

Вступ	4
Мета і завдання навчальної дисципліни	6
1. Класифікація і загальна характеристика інформаційних систем, які застосовують в системному аналізі	8
2. Системний аналіз як специфічний різновид системної діяльності	9
3. Інформаційне забезпечення системного аналізу	10
4. Основні методології проектування інформаційних систем	12
5. Методологія Agile виконання ІТ проєктів	14
6. Програмні засоби для управління виконанням ІТ проєктів	16
7. Структурний підхід до проектування інформаційних систем	18
8. Створення інформаційних систем за прецедентами	20
9. Ризики в ІТ проєктах та їх планування	22
10. Бізнес-аналітик у виконання ІТ проєктів	31
Завдання до лабораторних робіт	38
Список використаної літератури	39

Вступ

У сучасному світі інформаційні технології (ІТ) та системи стали невід'ємною частиною кожної сфери життя — від особистого спілкування до глобальної економіки. Знання в цій галузі відкриває широкі можливості для професійного розвитку, сприяє підвищенню ефективності роботи та забезпечує конкурентоспроможність у світі, який постійно змінюється.

Інформаційні технології значно оптимізують робочі процеси у всіх галузях. Володіння ІТ-інструментами, такими як програмне забезпечення для аналізу даних, автоматизації бізнес-процесів чи комунікації, дозволяє суттєво підвищити продуктивність праці, економити час і ресурси.

Знання інформаційних систем відкриває доступ до розробки нових продуктів, послуг і бізнес-моделей. ІТ-системи стають платформою для впровадження інновацій, таких як штучний інтелект, Інтернет речей (IoT) чи великі дані (Big Data), які змінюють підхід до управління, маркетингу, виробництва та інших сфер.

На сучасному ринку праці знання інформаційних технологій є однією з найважливіших навичок. Роботодавці високо цінують уміння працювати з сучасними системами, розуміння принципів кібербезпеки, а також здатність адаптуватися до нових технологій.

ІТ-технології не лише змінюють роботу, але й роблять повсякденне життя комфортнішим. Знання, як користуватися інформаційними системами, дозволяє спростувати виконання рутинних завдань, швидше знаходити інформацію, планувати час і забезпечувати власну кібербезпеку.

Інформаційні системи створюють основу для глобального обміну знаннями, співпраці між країнами та вирішення складних світових проблем. Ефективне використання таких систем сприяє сталому розвитку, покращенню екологічних процесів і впровадженню інновацій у медицину, освіту та інші критично важливі сфери.

Світ швидко змінюється, і цифрова трансформація охоплює все більше сфер. Знання інформаційних технологій і систем дозволяє

адаптуватися до цих змін, працювати ефективно у цифровому середовищі та бути частиною прогресу.

Таким чином, інформаційні технології та системи є ключовими інструментами для побудови успішного майбутнього. Їхнє розуміння дає змогу не лише залишатися конкурентоспроможним, а й робити свій внесок у розвиток суспільства й підвищення якості життя.

Мета і завдання навчальної дисципліни

Мета навчальної дисципліни полягає у формування у студентів теоретичних знань та практичних навичок використання сучасних інформаційних систем і технологій для аналізу, моделювання та підтримки прийняття рішень у складних системах різних предметних областей, а також для ефективного розв'язання задач системного аналізу.

Дисципліна формує такі компетентності:

ЗК4. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК5. Здатність генерувати нові ідеї (креативність).

ЗК6. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності).

ЗК7. Здатність виявляти та вирішувати проблеми на основі обґрунтованих рішень.

ЗК8. Здатність розробляти проекти та управляти ними.

СК8. Здатність розробляти і реалізовувати наукові та прикладні проекти в галузі інформаційних технологій та дотичні до неї між-дисциплінарні проекти.

Наведені результати навчання за відповідною дисципліною співвідносяться із *такими програмними результатами навчання*:

РН7. Розробляти інтелектуальні системи в умовах слабо структурованих даних різної природи.

РН13. Здатність ефективно здійснювати вибір концептуальної моделі середовища розподілених систем баз даних та знань на основі структурного та об'єктно-орієнтованого підходів.

1. Класифікація і загальна характеристика інформаційних систем, які застосовують в системному аналізі

Системний аналіз є важливою складовою сучасного управління, що охоплює дослідження та моделювання складних систем для оптимізації їх роботи. Інформаційні системи (ІС) відіграють ключову роль у цьому процесі, надаючи необхідні інструменти для збору, обробки та аналізу даних. Класифікація інформаційних систем, які використовуються в системному аналізі, допомагає краще зрозуміти їх функціональні можливості та області застосування.

Розглянемо класифікацію ІС з різними ознаками.

1. За рівнем інтеграції:

- *Автономні ІС*: працюють із даними в межах окремих підсистем, не інтегруючись з іншими системами.

- *Інтегровані ІС*: забезпечують обмін інформацією між різними підсистемами, що дозволяє створювати єдину інформаційну базу.

2. За функціональним призначенням:

- *Операційні системи*: підтримують повсякденні операції (наприклад, системи управління базами даних).

- *Аналізуючі системи*: призначені для збору та аналізу даних (наприклад, системи підтримки прийняття рішень).

- *Системи для моделювання*: використовуються для створення моделей реальних процесів з метою їх оптимізації.

3. За рівнем автоматизації:

- *Частково автоматизовані системи*: вимагають людської участі в процесах обробки даних.

- *Повністю автоматизовані системи*: виконують всі етапи обробки даних без участі людини.

4. За типом оброблюваних даних:

- *Дані про процеси*: інформація, що стосується операцій і бізнес-процесів.

-*Дані про ресурси*: інформація про матеріальні та нематеріальні ресурси підприємства.

-*Дані про середовище*: інформація, що стосується зовнішніх факторів, які можуть вплинути на діяльність системи.

Загальна характеристика інформаційних систем

Інформаційні системи, що застосовуються в системному аналізі, мають кілька ключових характеристик:

- *Модульність*: більшість сучасних ІС складаються з модулів, що дозволяє легко їх адаптувати під конкретні потреби користувачів.

- *Гнучкість*: здатність системи швидко адаптуватися до змін у бізнес-процесах або зовнішньому середовищі.

- *Інтерфейс*: зручний і інтуїтивно зрозумілий інтерфейс, що полегшує взаємодію користувача з системою.

- *Аналіз даних*: потужні аналітичні інструменти для обробки великих обсягів даних, що дозволяють отримувати корисну інформацію для прийняття рішень.

- *Безпека*: системи повинні мати належні засоби для захисту інформації від несанкціонованого доступу та витоків.

Інформаційні системи, що застосовуються в системному аналізі, є важливими інструментами для ефективного управління та оптимізації процесів. Їх класифікація за різними ознаками дозволяє краще зрозуміти їх функціональні можливості та вибрати найбільш підходящі рішення для конкретних задач. В умовах стрімкого розвитку технологій та зростання обсягів даних, значення таких систем тільки зростає, що підкреслює їх важливість у сучасному бізнес-середовищі.

2. Системний аналіз як специфічний різновид системної діяльності

Системний аналіз — це комплексний підхід, що застосовується для вивчення складних систем. Він базується на системному підході, що охоплює різноманітні аспекти досліджуваної проблеми. Системний аналіз допомагає зрозуміти структуру, функціонування та взаємозв'язки елементів системи, що робить його важливим інструментом у багатьох сферах, від бізнесу до екології.

Наведемо основні характеристики системного аналізу:

1. Комплексність: Системний аналіз розглядає систему в цілому, враховуючи всі її компоненти, зв'язки між ними та їхнє функціонування в різних умовах. Це дозволяє виявити неочевидні проблеми та потенційні можливості.
2. Моделювання: Однією з основних методик системного аналізу є створення моделей, які імітують поведінку реальних систем. Моделі можуть бути математичними, комп'ютерними або графічними і слугують для аналізу можливих сценаріїв розвитку.
3. Узгодженість: Системний аналіз передбачає узгодження всіх елементів системи, що дозволяє досягти ефективності у її функціонуванні. Це особливо важливо в складних системах, де навіть незначні зміни можуть призвести до суттєвих наслідків.
4. Ітеративність: Процес системного аналізу часто є ітеративним, що означає, що після початкового аналізу результати можуть бути переглянуті та уточнені на основі нових даних або змін у системі.

Системний аналіз включає різноманітні методи, такі як:

- *SWOT-аналіз*: оцінка сильних і слабких сторін, можливостей та загроз, що стоять перед системою.
- *Бізнес-моделювання*: створення моделей бізнес-процесів для оптимізації діяльності підприємства.
- *Системна динаміка*: аналіз поведінки складних систем у часі, вивчення їхньої стійкості та чутливості до зовнішніх впливів.
- *Аналіз чутливості*: вивчення, як зміни в одному з параметрів впливають на результати системи.

Системний аналіз знаходить своє застосування у багатьох сферах:

- Бізнес: для оптимізації бізнес-процесів, розробки стратегій і прийняття рішень.
- Інформатика: у розробці програмного забезпечення та управлінні проектами.
- Екологія: для вивчення екосистем і оцінки впливу антропогенної діяльності на навколишнє середовище.
- Соціологія: в аналізі соціальних систем та вивченні взаємозв'язків між різними соціальними групами.

Отже, системний аналіз є специфічним різновидом системної діяльності, що надає можливість глибоко вивчати та оптимізувати складні системи. Його важливість зростає в умовах швидко змінюваного світу, де ефективне управління та прийняття рішень стають критично важливими. Використання системного аналізу дозволяє організаціям адаптуватися до нових викликів і досягати своїх цілей більш ефективно.

3. Інформаційне забезпечення системного аналізу

Інформаційне забезпечення системного аналізу є ключовим аспектом, що визначає ефективність цього процесу. Воно охоплює всі етапи збору, обробки, зберігання та аналізу інформації, необхідної для прийняття обґрунтованих рішень. У світі, де дані стають найціннішим ресурсом, якість і доступність інформації стають критично важливими для успіху системного аналізу.

До основних компонентів інформаційного забезпечення можна віднести наступні.

1. Джерела інформації :

- Внутрішні джерела : дані, що збираються в рамках організації, такі як фінансові звіти, операційна інформація та статистика про продажі.

- Зовнішні джерела: інформація, що надходить ззовні, включаючи ринкові дослідження, статистику галузі, наукові публікації та аналітичні звіти.

2. Інформаційні системи :

- Системи управління базами даних (СУБД) : забезпечують зберігання та організацію даних, що дозволяє ефективно їх обробляти.

- Системи підтримки прийняття рішень (СППР) : використовують аналітичні інструменти для обробки даних і надання рекомендацій для прийняття рішень.

3. Методи збору та обробки інформації :

- Опитування та інтерв'ю : дозволяють отримати якісну інформацію від експертів та зацікавлених сторін.

- Аналіз даних : використання статистичних та математичних методів для обробки даних, виявлення трендів і закономірностей.

4. Моделювання :

- Використання моделей для імітації поведінки системи, що дозволяє прогнозувати наслідки різних рішень і стратегій.

Інформаційне забезпечення для системного аналізу дозволяє покращити

1. Якість прийнятих рішень: наявність точних і актуальних даних сприяє прийняттю обґрунтованих рішень, що базуються на фактах, а не на інтуїції.

2. Ефективність процесу: автоматизація збору та обробки інформації скорочує час, необхідний для аналізу, і дозволяє фахівцям зосередитися на критичних аспектах системи.

3. Гнучкість і адаптивність: системи, що постійно оновлюють свою інформаційну базу, здатні швидше реагувати на зміни у зовнішньому середовищі, що є критично важливим у сучасному бізнесі.

4. Співпрацю та комунікацію: ефективне інформаційне забезпечення забезпечує спільний доступ до даних усім учасникам процесу системного аналізу, що сприяє кращій комунікації та взаємодії.

Отже, інформаційне забезпечення системного аналізу є основою, на якій будується ефективність і точність аналізу. Якісні джерела інформації, сучасні інформаційні системи та методи збору та обробки даних створюють необхідні умови для прийняття обґрунтованих рішень. У сучасному світі, де дані набувають все більшої цінності, інвестиції в інформаційне забезпечення системного аналізу стають важливими для досягнення конкурентних переваг.

4. Основні методології проектування інформаційних систем

Проектування інформаційних систем (ІС) є складним і багатогранним процесом, що вимагає чіткого підходу та структури. Існує безліч методологій, які допомагають розробникам та аналітикам створювати ефективні, надійні та зручні у використанні системи. Розглянемо основні методології проектування інформаційних систем, їх характеристики та переваги.

1. Водоспадна модель

Водоспадна модель є класичним підходом до проектування ІС, в якому процес розробки розділено на послідовні етапи: аналіз вимог, проектування, реалізація, тестування, впровадження та обслуговування.

Переваги такої моделі у чіткій структурі і послідовності етапів, легкості у плануванні та контролі виконання та вона добре підходить для проектів із стабільними вимогами.

Недоліки цієї моделі у відсутності гнучкості, тобто важко внести зміни на пізніших етапах та вона не підходить для динамічних проектів, де вимоги можуть змінюватися.

2. Agile-методології

Методика Agile — це підхід, що акцентує увагу на гнучкості та адаптивності. Основна ідея полягає у розбитті проекту на короткі ітерації, що дозволяє постійно отримувати зворотний зв'язок від користувачів і швидко адаптувати систему до змін.

Переваги цієї методики у високій гнучкості у відповідь на зміни вимог, постійному зворотному зв'язку від замовника та користувачів та швидкому впровадженні нових функцій.

Недоліки цієї методики полягають у складності оцінки загального обсягу проєкту на початкових етапах та потребі тісної співпраці між командами.

3. Модель прототипування

У цій методології спочатку створюється прототип системи, що дозволяє замовнику отримати уявлення про кінцевий продукт. Прототипи можуть бути простими або детальними, в залежності від етапу розробки.

Переваги:

- Дозволяє швидко виявити та виправити помилки.
- Користувачі можуть краще зрозуміти вимоги до системи.
- Зменшує ризик невідповідності кінцевого продукту очікуванням замовника.

Недоліки:

- Може бути витратним за часом і ресурсами.
- Прототип може створити неправильне враження про функціональність.

4. Модель спіралеподібного проєктування

Ця модель поєднує елементи водоспадної моделі та прототипування. Проєкт розбивається на цикли (спіралі), кожен з яких містить етапи аналізу ризиків, планування, розробки, тестування та оцінки.

Переваги:

- Зосередженість на управлінні ризиками.
- Можливість вносити зміни на кожному етапі.
- Підходить для великих та складних проєктів.

Недоліки:

- Складність у реалізації.
- Вимагає досвіду для ефективного управління ризиками.

5. Об'єктно-орієнтоване проєктування

Ця методологія базується на концепціях об'єктно-орієнтованого програмування (ООП). Основна увага приділяється створенню об'єктів, які містять дані та методи для їх обробки.

Переваги:

- Краще модульність і повторне використання коду.
- Спрощує обслуговування і масштабування системи.
- Відповідає сучасним вимогам програмування.

Недоліки:

- Може бути складніше для нових розробників, які не знайомі з ООП.

- Вимагає значних зусиль на етапі проєктування.

Вибір методології проєктування інформаційних систем залежить від специфіки проєкту, вимог замовника та ресурсів, які є в наявності. Кожна з методологій має свої переваги і недоліки, і найчастіше успішне проєктування полягає у правильному поєднанні різних підходів для досягнення найкращих результатів. Важливо пам'ятати, що адаптація методологій до конкретних умов і потреб — це запорука успішного проєктування.

5. Методологія Agile виконання ІТ проєктів

Методологія Agile стала однією з найпопулярніших підходів до управління ІТ проєктами завдяки своїй гнучкості, адаптивності та орієнтації на потреби замовника. Основна мета Agile — забезпечити швидке реагування на зміни, ефективну комунікацію між командами та створення якісного продукту.

Основні принципи методології Agile

1. Ітеративність та інкрементальність. Проєкти виконуються через серію ітерацій (спринтів), кожна з яких має чіткі цілі та результати. Кожен спринт завершується робочим продуктом, що дозволяє замовнику отримувати регулярний зворотний зв'язок.

2. Співпраця. Agile акцентує увагу на тісній співпраці між командами, замовниками та користувачами. Регулярні зустрічі, такі як стендап-мітинги, дозволяють всім учасникам бути в курсі прогресу проекту.
3. Гнучкість у реагуванні на зміни. Agile визнає, що вимоги можуть змінюватися протягом життя проекту. Замість того, щоб намагатися їх зафіксувати на початку, команда гнучко реагує на нові вимоги і адаптує план.
4. Фокус на кінцевому користувачеві. Основна мета Agile — створити цінність для користувачів. Регулярний зворотний зв'язок допомагає визначити, чи відповідає продукт очікуванням.

Основні методи Agile

1. *Scrum*. Це одна з найпопулярніших фреймворків Agile, що використовує короткі цикли (спринти) для виконання завдань. Scrum передбачає наявність ролей (власник продукту, Scrum-майстер, команда розробників), а також регулярних зустрічей (планування спринту, щоденні стендап-мітинги, огляд спринту та ретроспектива).
 2. *Kanban*. Kanban акцентує увагу на візуалізації процесу роботи. Використовуючи канбан-дошки, команди можуть легко відстежувати прогрес завдань, що дозволяє оптимізувати робочий процес і зменшити затримки.
 3. *Extreme Programming (XP)*. Цей підхід фокусується на поліпшенні якості програмного забезпечення через часті випуски, парне програмування, тестування і регулярні зворотні зв'язки.
- Переваги Agile очевидні, але вкажемо основні
- *Швидка доставка продукту*. Завдяки ітеративному підходу команди можуть швидко випускати працюючі версії продукту.
 - *Зменшення ризиків*. Постійний зворотний зв'язок дозволяє рано виявляти проблеми та адаптуватися до змін.
 - *Підвищена мотивація команди*. Гнучкість і можливість впливати на проект підвищують залученість членів команди.
 - *Вища якість продукту*. Постійне тестування і зворотний зв'язок забезпечують якість кінцевого продукту.

Виклики Agile

- Необхідність досвіду. Успішне впровадження Agile вимагає досвіду команди та розуміння принципів методології.
- Проблеми з масштабуванням. У великих проектах може виникати складність у координації між багатьма командами.
- Зміна культури. Впровадження Agile може вимагати значних змін у корпоративній культурі та процесах.

Методологія Agile пропонує ефективний підхід до виконання ІТ проектів, що дозволяє адаптуватися до змін, підтримувати тісну співпрацю між усіма учасниками та створювати цінні продукти. Хоча Agile має свої виклики, правильне впровадження може суттєво покращити результати проекту та задоволеність клієнтів.

6. Програмні засоби для управління виконанням ІТ проектів

Управління ІТ проектами вимагає використання різноманітних програмних засобів, які допомагають організувати процеси, контролювати виконання завдань, комунікувати в команді та аналізувати результати. Ось деякі з найбільш поширених категорій та прикладів програмних засобів, що використовуються в управлінні ІТ проектами.

6.1. Системи управління проектами

Ці програми допомагають планувати, організовувати, контролювати і завершувати проекти.

- Microsoft Project. Один з найбільш відомих інструментів, який дозволяє створювати графіки проектів, відстежувати виконання завдань, а також управляти ресурсами.
- Smartsheet. Гнучка платформа, що поєднує в собі функції таблиць і управління проектами, дозволяючи команді легко співпрацювати та відстежувати прогрес.

6.2. Інструменти для Agile-управління

Ці програми оптимізують управління проектами в Agile-середовищі, підтримуючи ітеративний підхід.

- Jira. Популярний інструмент для управління проектами в Agile, що дозволяє планувати спринти, відстежувати завдання та вести облік проблем.
- Trello. Візуальний інструмент для управління завданнями, який використовує канбан-дошки для організації робочого процесу.

6.3. Інструменти для комунікації та співпраці

Ефективна комунікація є критично важливою для успіху проекту, і ці інструменти допомагають командам залишатися на зв'язку.

- Slack. Платформа для командної комунікації, що дозволяє обмінюватися повідомленнями, файлами та інтегрувати інші інструменти.
- Microsoft Teams. Платформа для спільної роботи, яка об'єднує функції чату, відеоконференцій та обміну документами.

6.4. Інструменти для управління ресурсами

Ці програми допомагають організувати та оптимізувати використання ресурсів у проектах.

- Resource Guru. Інструмент для планування ресурсів, який дозволяє відстежувати доступність та навантаження команди.
- 10,000ft. Платформа, що пропонує прості у використанні інструменти для планування ресурсів та управління проектами.

6.5. Інструменти для аналізу даних та звітності

Для моніторингу прогресу проекту важливо мати можливість аналізувати дані і генерувати звіти.

- Tableau. Інструмент для візуалізації даних, що дозволяє створювати інтерактивні звіти та дашборди для відстеження показників проекту.
- Power BI. Аналітичний інструмент від Microsoft, який забезпечує можливості для створення звітів та візуалізації даних.

6.6. Інструменти для управління ризиками

Ці програми допомагають ідентифікувати, аналізувати та управляти ризиками в проєктах.

- RiskWatch. Платформа для управління ризиками, що дозволяє оцінювати, відстежувати та управляти ризиками в проєкті.

- @RISK. Інструмент, що працює з Excel, який допомагає аналізувати ризики та невизначеність в проєктах.

Використання програмних засобів для управління виконанням ІТ проєктів є необхідним для підвищення ефективності, контролю за прогресом та забезпечення успішного завершення проєкту. Вибір конкретних інструментів залежить від потреб команди, специфіки проєкту та методології, що застосовується. З правильними програмними засобами управління проєктами можна досягти більшої продуктивності, зменшити ризики та покращити загальний результат.

7. Структурний підхід до проєктування інформаційних систем

Структурний підхід до проєктування інформаційних систем (ІС) є одним із основних методів, що дозволяє систематично аналізувати, проєктувати та реалізовувати складні інформаційні системи. Цей підхід акцентує увагу на чіткій структурі і послідовності етапів розробки, що допомагає забезпечити високий рівень організації та контролю над проєктом.

7.1. Основні принципи структурного підходу

1. Чітка структура. Структурний підхід базується на розділенні системи на менші, управлінські частини або модулі. Це дозволяє зосередитися на кожному компоненті окремо, полегшуючи процес аналізу та розробки.

2. Орієнтація на дані. Важливе місце в структурному підході займає аналіз даних, що обробляються системою. Це передбачає визначення структури даних, їх взаємозв'язків і способів обробки.

3. Процесний підхід. Структурний підхід включає в себе також детальне описання процесів, що виконуються в системі. Це дозволяє зрозуміти, як дані переміщуються між різними модулями та які операції виконуються.

4. Узгодженість. Всі елементи системи повинні бути взаємопов'язаними і відповідати загальним вимогам. Це забезпечує цілісність і узгодженість у розробці.

7.2. Етапи структурного підходу

1. Аналіз вимог. На цьому етапі визначаються основні вимоги до системи, її функціональні можливості, а також потреби кінцевих користувачів. Використовуються різноманітні методи, такі як інтерв'ю, опитування та спостереження.

2. Проектування. Основна мета цього етапу — створити детальний план системи. Він включає в себе розробку діаграм, які відображають структуру даних, процеси та взаємозв'язки між компонентами.

3. Реалізація. На цьому етапі здійснюється програмування та розробка компонентів системи відповідно до попереднього проекту. Важливо забезпечити належну документацію для кожного модуля.

4. Тестування. Після реалізації проводиться тестування системи для виявлення помилок та нес відповідностей. Це включає в себе як модульне тестування, так і інтеграційне тестування.

5. Впровадження. На цьому етапі система вводиться в експлуатацію, і користувачі отримують необхідні навчальні матеріали та підтримку для ефективного використання.

6. Обслуговування. Після впровадження система потребує регулярного обслуговування, що включає в себе оновлення, виправлення помилок та вдосконалення функцій.

7.3. Переваги структурного підходу

- Системність. Чітка структура дозволяє зрозуміти, як різні компоненти системи взаємодіють між собою.

- Прозорість. Кожен етап має свої цілі і завдання, що забезпечує прозорість процесу розробки.

- Контроль. Завдяки структуризації та чітким етапам проекту легше контролювати хід виконання і виявляти відхилення від плану.

- Повторне використання. Модульність системи дозволяє повторно використовувати окремі компоненти в інших проектах.

Структурний підхід до проектування інформаційних систем є ефективним методом, який дозволяє забезпечити високий рівень організації, контролю та якості в розробці. Його принципи і етапи допомагають створювати системи, що відповідають потребам користувачів і мають високу ступінь надійності. Завдяки своїй системності та ясності, структурний підхід залишається популярним серед фахівців у сфері ІТ.

8. Створення інформаційних систем за прецедентами

Створення інформаційних систем за прецедентами є важливим підходом у сфері програмної інженерії, який дозволяє розробникам використовувати досвід попередніх проектів для ефективного вирішення завдань у нових системах. Цей метод передбачає вивчення та аналіз уже реалізованих рішень, що дозволяє запобігти помилкам, скоротити час на розробку та підвищити якість кінцевого продукту.

Основні принципи створення інформаційних систем за прецедентами полягають у наступному.

1. Вивчення прецедентів. Першим кроком є ідентифікація та аналіз успішних (і неуспішних) проектів, які мають подібні характеристики або вимоги. Це може включати огляд документації, інтерв'ю з учасниками проектів, а також вивчення готових рішень.

2. Адаптація рішень. Важливо не просто копіювати готові рішення, а адаптувати їх до специфічних умов і вимог нової системи. Це може включати зміну архітектури, інтерфейсу користувача або технологічного стеку.

3. Документація. Ведення детальної документації про прецеденти, які були використані в проекті, допомагає команді зберігати знання та досвід, що може бути корисним для майбутніх проектів.

4. Модульність. Використання модульних компонентів, які вже були протестовані в інших проектах, дозволяє прискорити розробку та зменшити ризики, пов'язані з новими рішеннями.

Переваги використання прецедентів

- Зменшення ризиків : Використання перевірених рішень допомагає уникнути типових помилок, що можуть виникнути в нових проектах.
- Скорочення термінів розробки : Наявність готових компонентів або рішень дозволяє швидше завершувати етапи розробки.
- Покращення якості : Застосування вже відпрацьованих рішень підвищує надійність і якість інформаційної системи.

Основні виклики при використанні прецедентів

- Необхідність адаптації : Готові рішення не завжди можуть бути використані в первозданному вигляді. Це вимагає додаткових зусиль на їх адаптацію до нових вимог.
- Обмеженість прикладів : Якщо в галузі недостатньо прецедентів, це може ускладнити процес розробки.
- Ризик залежності : Існує ймовірність, що надмірне використання готових рішень призведе до втрати інноваційності та адаптивності.

Створення інформаційних систем за прецедентами є ефективним підходом, що дозволяє використовувати накопичений досвід для підвищення якості та зменшення ризиків у процесі розробки. Незважаючи на певні виклики, цей метод може значно спростити процес проектування та реалізації нових систем. Використання прецедентів не тільки полегшує роботу розробників, але й сприяє створенню більш стабільних та ефективних рішень, що відповідають потребам користувачів.

9. Ризики в ІТ проєктах та їх планування

Ризики – непередбачувані обставини або події, які можуть вплинути на проєкт як позитивно, так і негативно. Область впливу ризиків не є обмеженою, тому вони можуть “ставити свій відбиток” на якості та ефективності роботи людських та технологічних ресурсів, процесів тощо. Сфери походження ризиків також можуть відрізнятися, наприклад: технологічна, економічна, соціальна, політична сфери, природні фактори та ін.

Усі ризики характеризуються такими факторами:

- Суть ризику. Інформація про те, яка подія може статися і вплинути на хід розробки проєкту;
- Часові межі. Ймовірні часові проміжки, коли ризик може настати;
- Імовірність. Оцінка того, які шанси настання того чи іншого ризику;
- Вплив. Характеристика наслідків, які можуть бути після настання ризиків;
- Тригери. Інформація про події, які можуть викликати ризики.

Розглянемо типи ризиків, що є важливими для ІТ-проєктів:

1. Технічні ризики. Сюди входять технічні збої, системні помилки та порушення, кібератаки, проблеми із сумісністю та ін.;
2. Фінансові ризики. Характеризуються необхідністю у більшій кількості коштів, аніж було заплановано. Це є результатом неточного збору вимог та планування, що призводить до незадоволення клієнта, або до неможливості успішного завершення проєкту;
3. Ризики, пов’язані із графіком робіт. Неправильне планування призводить не тільки до додаткових фінансових затрат, а й до зсуення графіку робіт, і, як наслідок, пропуску запланованих дедлайнів;
4. Ризики, пов’язані із результатом. Цей тип ризиків є областю відповідальності і сторони виконавця, і сторони замовника. Велику роль для цього типу ризиків грає вміння пояснювати та конкретизувати – з боку клієнта, та збирати і узгоджувати – з

боку розробника, вимоги до програмного продукту. Але варто зазначити, що жоден із учасників проєкту не може передбачити чи користуватиметься популярністю кінцевий продукт, і чи буде він актуальним на момент релізу;

5. Організаційні ризики. Тригером для цієї групи ризиків є низька якість організації роботи, неправильний поділ та призначення відповідальних осіб для проєктних задач, погано продумані плани закупівлі та розподілу ресурсів тощо;
6. Комунікаційні ризики. Найбільш поширеними ризиками цієї категорії є поява непорозумінь, відсутність зв'язку між учасниками проєкту та ін., що можуть бути викликаними через недостатню чіткість і ясність вираження ідей та думок членами команди, мовні бар'єри, культурні бар'єри тощо;
7. Ризики, пов'язані із обсягами робіт. Часто у замовників виникають нові ідеї, потреби і бажання по ходу розробки продукту, або ж змінюються потреби бізнесу чи ринку, тому виникає необхідність у тому, аби виходити за межі початково встановленого обсягу робіт. Наслідком цього ризику можуть бути збільшення фінансових затрат, зсув графіку, зниження якості продукту тощо;
8. Ризики, пов'язані із навичками команди розробників. Різноманітні зміни, виникнення технічних проблем та інших факторів, можуть призвести до того, що команда не буде містити в собі достатньо кваліфікованих та досвідчених кадрів для вирішення відповідного питання.

Для глибшого розуміння, наведемо приклади ризиків, які можуть настати під час роботи над програмним продуктом:

- Збій або несправність програмно-технічних засобів;
- Шпигунські та шкідливі програми. Це означає захоплення систем за допомогою даних програм, викрадення конфіденційної інформації або її використання проти компанії-власника проєкту;
- Хакерство. Злом системи контролю, а отже отримання хакером доступу до усіх файлів та важливої інформації;

- Стихійні лиха;
- Незадоволені потреби клієнта. Цей ризик може бути зумовленим різними причинами: непрофесіоналізм компанії-розробника, застарілість технічних навичок її робітників, неякісний збір вимог та потреб замовника тощо;
- Втрата даних. Якщо система є вразливою, то наслідком може стати утрата даних через нестабільність системи, або крадіжка даних. Даний ризик несе за собою тяжкі наслідки у вигляді штрафів або ж відкриття юридичних дій різного характеру в залежності від конкретного випадку;
- Втрачені можливості. Причиною цього ризику є відсутність моніторингу змін на ринку та технологічних трендів з боку компанії-виконавця, що призводить до неактуальності програмного продукту або його застарілості.

Отже, бачимо, що існує велика кількість ризиків, які можуть настати під час роботи над проектом. Усі вони різняться за джерелом виникнення, ступенем впливу та методами вирішення, але повинні бути детально розглянуті для успішного ведення та завершення проекту. Тому виникає необхідність у менеджменті проектних ризиків, що є наступним предметом розгляду.

Управління ризиками – це процес виявлення, аналізу, попередження та реагування на ризики, що виникають протягом життєвого циклу проекту. Основним завданням процесу управління ризиками є якісне планування ризиків, аби попередити виникнення проблем, з якими, зазвичай, набагато важче боротися.

Управління ризиками є необхідним і корисним не лише у виявленні та усуненні негативних ризиків, але й в ідентифікації позитивних ризиків, тобто певних можливостей до покращення якості роботи, окремого продукту чи вартості бізнесу в загальному. Менеджмент ризиків передбачає вивчення взаємозв'язків між ризиками та їх каскадним впливом на хід проекту, тому варто складати та розглядати загальну “картину” системи ризиків та використовувати цілісний підхід при їх аналізі.

Як і кожен процес, управління ризиками проекту має визначені етапи, а саме:

1. Ідентифікація ризиків. Метою цього етапу є визначення усіх можливих ризиків, що можуть виникнути під час життєвого циклу проекту;
2. Аналіз ризиків та оцінка ймовірності їх настання. Визначення джерел ризиків, значимості їх наслідків, оцінки імовірності їх настання та ступінь впливу на проект;
3. Пріоритезація ризиків, базуючись на бізнес-цілях. До даної фази входять планування ризиків, розробка плану управління ними для того аби їм запобігти або зменшити їх вплив;
4. Реагування на ризики. На цьому етапі план управління ризиками починає діяти, тому реалізуються усі заплановані заходи щодо менеджменту ризиків;
5. Моніторинг та адаптація. Хоча цей етап наведений останнім у списку фаз менеджменту ризиків проекту, він не є завершальним. Моніторинг є обов'язковим та постійним процесом під час управління ризиками. Необхідним є відслідковування та контролювання ризиків протягом усього процесу роботи над програмним продуктом, аби могли вчасно та якісно приймати певні рішення і заходи щодо усунення або запобігання ризиків.

Для того, аби могли ефективно вести та управляти усіма перерахованими етапами менеджменту ризиків в проекті, необхідно вести відповідну проектну документацію, що дозволить якісно збирати, аналізувати та зберігати важливу інформацію. Розглянемо артефакти, пов'язані із управлінням ризиками:

- Беклог, із урахуванням ризиків (Risk-adjusted backlog). Містить обсяги робіт із урахуванням роботи та певних дій над усуненням загроз та використанням можливостей;
- Реєстр ризиків (Risk register). Включає в себе загальний опис ризику, його оцінку, міру впливу та відповідальну особу за менеджмент відповідного ризику;

- План управління ризиками. Характеристика дій для запобігання або плану усунення того чи іншого ризику, структуризація процесів боротьби із ними;
- Структурне розбиття ризиків (Risk breakdown structure). Даний артефакт має вигляд діаграми, що являє собою ієрархічне представлення потенційних джерел ризиків;
- Звіт про ризики (Risk report). Цей документ ведеться протягом усього життєвого циклу проекту і підводить підсумки щодо менеджменту ризиків проекту, а також щодо рівня загального проектного ризику.

Отже, можемо зробити висновок, що документація менеджменту ризиків проекту є необхідною для якісного та ефективного виконання даного процесу, і грає ключову роль у моніторингу і контролі проектних ризиків.

Для того, аби мати можливість управляти ризиками, спершу потрібно вміти їх виявляти та аналізувати. Найперший крок, потрібний для ідентифікації ризиків – збір інформації про проєкт. Сюди входить інформація про проєктну ціль, обсяги робіт та обмеження у вигляді часу, бюджету, ресурсів тощо. Також важливо визначити якомога більше зацікавлених сторін – стейкхолдерів, які можуть тим чи іншим чином впливати на хід розвитку проєкту. Дуже корисним фактором у ідентифікації ризиків є використання досвіду із минулих проєктів командою розробників, оскільки деякі проєкти можуть мати схожі ризики або взагалі ідентичні, тому компанія-виконавець може застосувати вже розроблені техніки усунення чи запобігання ризикам, які забезпечили успішний менеджмент ризиків проєкту в попередній роботі. Не менш важливим є консультативний процес із спеціалістами конкретної галузі, для якої розробляється програмне забезпечення, а також загальні збори команди для брейнштормінгу і генерації нових ідей щодо виявлення, запобігання та усунення ризиків.

Для аналізу ризиків в IT-проєктах найчастіше використовуються такі моделі аналізу, як: SWOT-аналіз та PESTLE-аналіз. Розглянемо кожен із них більш детально.

SWOT-аналіз - інструмент для виявлення внутрішніх слабких сторін проєкту та оцінки внутрішніх загроз та можливостей.

- S – Strength. Сильні внутрішні сторони, позитивні атрибути проєкту – технології, фінанси, високий рівень навичок команди, знання тощо;
- W – Weakness. Слабкі внутрішні сторони, тобто певні негативні фактори, які потрібно викоринити або вдосконалити для більш ефективної роботи над проєктом;
- O – Opportunity. Можливості – зовнішні позитивні фактори, які можуть посприяти успіху проєкту;
- T – Threat. Загрози – зовнішні фактори, на які не можна вплинути. Потрібно розробити план дій для випадку справдження цих обставин.

PESTLE-аналіз - інструмент для виявлення зовнішніх факторів, які можуть вплинути на проєкт та процес його планування.

- P – Political. Політичні фактори, прикладом можуть слугувати податкова політика, фіскальна політика, торговельні штрафи та ін.;
- E – Economical. Економічні фактори, до яких належать інфляція, відсоткові ставки, валютні курси тощо;
- S – Social. Соціальні фактори, предметом уваги яких є культурні тенденції, демографія, аналітика населення тощо;
- T – Technological. Технологічні фактори стосуються появи нових технологій, інноваційної діяльності на ринку, різноманітних досліджень і розробок;
- L – Legal. Юридичні фактори включають закони, які можуть вплинути на проєкт тим чи іншим чином;
- E – Environmental. Екологічні фактори – це усі фактори, що пов'язані із навколишнім середовищем, і зазвичай обмежені такими складовими, як клімат, погода, географічне положення та ін.

Отже, SWOT-аналіз та PESTLE-аналіз є важливими моделями для аналізу ризиків проєкту, і допомагають у виявленні та оцінюванні як внутрішніх, так і зовнішніх факторів, які мають вплив на проєкт.

1.1 Методи оцінювання ризиків: Методи Дельфі та матриці ймовірності

Як було зазначено раніше, процес ідентифікації, прогнозування та оцінки ризиків є надзвичайно важливим для того, аби мати змогу якісно ними керувати та ефективно вести розробку проекту. Розглянемо один із методів, що забезпечує даний процес, а саме метод Дельфі.

Метод Дельфі був створений у США на початку Холодної війни для того, аби дати його користувачам змогу прогнозувати вплив технологій на ведення війни. Тобто початковою ціллю даного методу був збір експертної інформації задля забезпечення процесу прогнозування наукових технологій, застосовуваних в ході війни.

Завдяки даному методу було відкрито нові перспективи оцінки ризиків, оскільки метод полягає у залученні експертів із різних областей, і, як наслідок, отримання загальної об'єктивної оцінки. Метод Дельфі дозволяє ідентифікувати, прогнозувати і оцінювати ризики більш обґрунтовано та об'єктивно, тому у команди проекту виникає можливість якісного планування та боротьби із ризиками, застосування ефективних заходів для зменшення їх впливу та наслідків. Метод Дельфі складається із таких основних етапів:

1. Визначення проблеми чи аспекту, який потребує оцінки ризиків;
2. Вибір експертів, які будуть брати участь у опитуванні;
3. Складання запитань, на які експертам необхідно дати відповідь;
4. Проведення сету анонімних опитувань за допомогою надсилення учасникам опитувальних форм. Анкетування може проводитись через мережу Інтернет, електронну пошту та інші засоби віртуального зв'язку;
5. Проведення аналізу результатів;
6. Формування фінального звіту. Підведення підсумків щодо проведеного процесу оцінки ризиків складанням звіту, що

містить відповіді експертів на запитання, аналіз результатів та рекомендації щодо подальшої роботи.

Як бачимо, жоден із етапів даного методу не передбачає дискусій та обговорень, а тому і виключає можливість необ'єктивного та непрозорого оцінювання, що спричинене впливом різних факторів. Обговорення, у даному випадку, замінені послідовними індивідуальними опитуваннями – анкетуванням. Із проведенням кожного наступного кола анкетування, експерти обґрунтовують та уточнюють свою точку зору, а також ознайомлюються із думками інших учасників опитування і порівнюють свою точку зору із узагальненими висновками групи учасників-експертів. Проведення опитувань проводиться доти, доки експерти не досягнуть консенсусу. Метод Дельфі має такі основні принципи:

- Анонімність. Експерти відповідають на анкетування анонімно, тому можуть вільно висловлювати свої прогнози та оцінки;
- Ітераційність. Проводиться декілька раундів анкетування, в кожному з яких учасники відповідають на питання, а потім отримують фідбек від інших учасників групи і мають змогу скоригувати свої прогнози щодо конкретного питання, і в наступному раунді надають їх знову;
- Статистична обробка результатів анкетування групи експертів;
- Структурований процес. Він забезпечує зосередження на основній темі опитування і проведення деталізованого аналізу ризиків аспекту.

Наведені принципи є також і вагомими перевагами даного методу прогнозування і оцінювання ризиків. Але, є і недоліки, а саме:

- Часові витрати. Процес анкетування може потребувати великих витрат часу, оскільки може виникати необхідність у проведенні додаткових турів анонімних опитувань, а також у вивченні та аналізі інформації та підведенні підсумків;
- Людський фактор. Іноді оцінки можуть бути необ'єктивними, оскільки деякі експерти є більш досвідченими та впевненими у своїй професійній думці, тому інші,

менш впевнені в собі експерти, можуть відступати та відмовлятися від своєї точки зору під впливом даного фактору;

- Ймовірність неправильного підбору експертів. Можливість хибного визначення поняття експерта для конкретного предмету анкетування, і, як наслідок, отримання спотворених та недостовірних даних.

Наразі метод Дельфі використовується в різних сферах діяльності, але не є універсальним рішенням і має як позитивні, так і негативні аспекти у застосуванні в окремих проєктах.

Метод побудови матриці ймовірності – метод, що застосовується для оцінки ризиків в управлінні проєктами, бізнесом тощо. Він допомагає оцінювати ймовірність настання ризику та наслідки його впливу на проєкт. Для побудови матриці ймовірності потрібно виконати наступні кроки:

1. Визначити ймовірні події та їх джерела;
2. Створити модель залежностей визначених ризиків;
3. Визначити ймовірності настання кожного ризику, попередньо скориставшись обраним методом оцінки, статистичними даними, досвідом із схожих минулих проєктів, експертними оцінками тощо;
4. Поділити події за рівнем ймовірності настання та рівнем впливу на проєкт;
5. Підготувати план із мінімізації настання та рівня наслідків визначених ймовірних подій.

Матриця ймовірності та впливу являє собою список групованих подій за певним рівнем оцінок, що були визначені та враховані раніше, використовуючи підходящий для конкретного проєкту метод. Ймовірності та рівні наслідків окремих ризиків формують категорії за діапазонами значень. Елементами матриці виступають значення добутків оцінок ймовірності настання подій та вагомості їх впливу на проєкт. Дані елементи складають матрицю, що поділена на сегменти в залежності від отриманих результатів. Приклад матриці ймовірності та впливу наведений у посібнику РМВОК, та відображений у таблиці, що наведена нижче.

Таблиця №4 – Матриця ймовірності та впливу

Імовірність	Загрози					Можливості				
0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09	0.05
0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04
0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03
0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02
0.10	0.01	0.01	0.02	0.04	0.08	0.08	0.04	0.02	0.01	0.01
	0.05	0.10	0.20	0.40	0.80	0.80	0.40	0.20	0.10	0.05

Із наведеної таблиці можемо спостерігати різні рівні ризиків, а саме відповідно до порогів: 0.05 – дуже низький, 0.10 – низький, 0.20 – середній, 0.40 – високий, 0.80 – дуже високий. Таким чином кожен оцінюється кожен ризик, і з’являється можливість вибору превентивних заходів для окремих ризиків, а також розробки планів управління та боротьби із ними.

Оцінка ризиків є ключовим етапом у управлінні проектами, що дозволяє виявляти, аналізувати та управляти потенційними загрозами, які можуть вплинути на досягнення цілей проекту. Зважаючи на складність і динамічність сучасного бізнес-середовища, ефективне управління ризиками стає необхідною умовою для успішної реалізації проектів.

10. Бізнес-аналітик у виконання ІТ проектів

Бізнес-аналітик (БА) — це фахівець, який збирає та виявляє бізнес-вимоги до майбутнього продукту/функціоналу, аналізує, передає їх команді розробки та сприяє реалізації обґрунтованих бізнес-рішень, які покращують ефективність та продуктивність компанії-замовника.

На проєктах, пов’язаних із розробкою програмного забезпечення, бізнес-аналітик відіграє ключову роль. Часто його називають “перекладачем з бізнесової на програмістську”, адже він має перетворити ідеї на бізнес-вимоги так, щоб не втратити цінну інформацію, а далі описати вимоги так, щоб було зрозуміло і команді розробників, і бізнесу.

З одного боку, цей фахівець повинен мати глибокі знання бізнесу, добре розуміти нішу замовника, його бізнес-потреби, обмеження, розуміти, які функції ПЗ задовольняють потреби кінцевих користувачів, та як їх реалізувати.

З іншого боку, бізнес-аналітик у сфері ІТ має добре розуміти процес розробки ПЗ, технічні тонкощі кожного етапу SDLC, розмовляти мовою технологій.

Завдання та функції бізнес-аналітика варіюються залежно від проєкту, розміру компанії та взагалі уявлення керівництва про роль БА.

Зазвичай завдання та обов'язки бізнес-аналітика включають:

- **Збір та аналіз вимог.** Бізнес-аналітик збирає, аналізує та документує бізнес-вимоги від замовника та стейкхолдерів. Він розробляє детальні специфікації, які будуть використовуватися ІТ-командою для створення продукту/функціоналу.
- **Аналіз ринку.** БА аналізує продукти конкурентів та розробляє стратегії для підвищення конкурентоспроможності компанії-замовника.
- **Моделювання бізнес-процесів.** Business Analyst моделює бізнес-процеси, використовуючи різні інструменти (наприклад, BPMN). Це допомагає виявити недоліки в процесах та розробити варіанти їх удосконалення.
- **Оцінка ризиків.** Бізнес-аналітик в ІТ прогнозує можливі ризики, пов'язані з розробкою та реалізацією продукту, та надає рекомендації щодо їх мінімізації.
- **Комунікація та координація.** Більша частина роботи БА — це комунікація із замовником та членами команди (програмістами, дизайнерами, тестувальниками, проєктними менеджерами). Він виступає як місток між бізнесом та технічними командами, забезпечуючи, щоб всі члени команди мали чітке розуміння цілей та вимог проєкту.
- **Управління вимогами та змінами.** На основі зібраних вимог аналітик створює документацію, де описує детально функціональні та нефункціональні вимоги до продукту. Бізнес-аналітик

відповідає за управління змінами на проєкті, включно з оновленням вимог і специфікацій.

- **Забезпечення якості.** БА збирає та аналізує зворотний зв'язок від учасників проєкту, проводить оцінку результатів.

Робота бізнес-аналітика в ІТ-компанії зазвичай складається з таких етапів:

- *Підготовка.* На цьому етапі відбувається зустріч із замовником, збір вихідних даних, визначення цілей проєкту. Бізнес-аналітик встановлює зв'язки з ключовими зацікавленими сторонами, визначає їхні потреби та очікування.

- *Збір вимог.* Інтерв'ювання зацікавлених осіб, опитування для збору всіх необхідних вимог до проєкту, ідентифікації проблем та визначення бізнес-потреб.

- *Аналіз та специфікація.* Business Analyst документує, аналізує, класифікує вимоги, встановлює їх пріоритет. Після цього розробляє детальні специфікації, в яких визначає необхідні артефакти, детально описує вимоги й далі передає їх команді.

- *Проектування рішень.* На цьому етапі бізнес-аналітик разом із командою працює над розробкою найкращих рішень для вирішення бізнес-задач: як краще спроектувати та впровадити рішення, які є технічні обмеження тощо.

- *Реалізація.* Бізнес-аналітик моніторить виконання проєкту, забезпечуючи, щоб усі аспекти відповідали специфікаціям та бізнес-вимогам.

- *Тестування та валідація.* Часто БА залучається до тестування та перевірки функціоналу, щоб перевірити відповідність розробленого рішення вимогам та бізнес-цілям.

- *Впровадження та підтримка.* Після запуску проєкту бізнес-аналітик може бути залучений до підтримки, збору зворотного зв'язку для подальшого удосконалення продукту.

На великих проєктах практикують залучення не тільки бізнес-аналітика, а й аналітика даних. Хоча бізнес-аналітик (Business Analyst) та аналітик даних (Data Analyst) можуть здаватися схожими професіями через їхню роботу з даними, насправді вони зосереджені на різних аспектах аналізу та виконують різні задачі.

Аналітик даних працює з даними, що надходять із різних джерел, тоді як бізнес-аналітик (БА) працює з вимогами до продукту.

Обов'язки бізнес аналітика на іт апроєкті:

- мета роботи — поліпшити бізнес-процеси, забезпечити впровадження ефективних системних рішень та максимізувати цінність для бізнесу через технології;
- робота спрямована на реалізацію проєктів від старту до запуску і подальшу підтримку;
- фокусується на з'ясуванні бізнес-вимог, їх аналізі та передачі цих вимог команді розробників, дизайнерів та інших учасників проєкту;
- за потреби — проводить аналіз схожих продуктів у конкурентів
- комунікує з усіма учасниками проєкту та стейкхолдерами;
- приймає та оцінює результати роботи: чи відповідають вони очікуванням та вимогам бізнесу;
- ключові хард скілли включають розуміння бізнес-процесів, управління проєктами, моделювання бізнес-процесів, також вміння користуватися інструментами збору та візуалізації даних (Excel, Google Sheets, SQL, Tableau, Power BI тощо).

Що робить аналітик даних?

Мета його роботи — допомогти бізнесу приймати рішення на основі даних, що включає ідентифікацію нових можливостей та оцінку ефективності поточних стратегій а також:

- робота зосереджена на аналізі та прогнозуванні, що забезпечує керівництво даними для стратегічних та оперативних рішень;
- збирає та обробляє дані з безлічі різноманітних джерел;
- аналізує та візуалізує дані у вигляді таблиць, графіків, діаграм, дашбордів;
- перевіряє гіпотези, проводить тестування;
- шукає пояснення подіям, причинно-наслідкові зв'язки, аномалії, збіги, що допомагає компаніям оперативно відповідати на різноманітні запитання (наприклад, чому впав N показник у N регіоні) та приймати виважені data-driven рішення;

- базові хард скілли аналітика даних — знання статистики, інструментів збору й обробки даних (Excel, Google Sheets, SQL), BI систем (Tableau, Power BI), мов Python та R.

Плюси та мінуси професії бізнес-аналітика

Як і в будь-якій професії, робота БА має свої плюси та мінуси.

Переваги професії:

- Business Analyst може працювати в різних доменах, компаніях, проєктах, виконувати різний спектр завдань, знаходити інноваційні шляхи розв'язання проблем, що робить його роботу цікавою та різноманітною;
- кар'єрні перспективи у бізнес аналітика в ІТ: ВА може рухатися далі в напрямку менеджменту (ставати ПМ, переходити у продакт менеджмент), стати Scrum-майстром або ж перейти в аналітику даних (наприклад, Big Data Analyst), BI (business intelligence) чи взагалі піти в розробку;
- професійне зростання: щоб бути затребуваним спеціалістом, БА має постійно вчитися, прокачувати свої скіли, що веде до його професійного розвитку;
- можливість бачити весь процес роботи над проєктом та впливати на всі його аспекти;
- високий попит на спеціалістів, адже вони допомагають бізнесу підвищувати ефективність роботи, виявляти проблеми та недоліки, які потенційно можуть зменшити прибутковість та задоволеність клієнтів.

Недоліки роботи бізнес-аналітиком:

- дуже висока відповідальність за ухвалення важливих рішень та успіх проєкту;
- замовники не завжди надають точні та чіткі вимоги, тому можливі недоліки в якості, затримки проєкту, втрата коштів;
- роль бізнес-аналітика в ІТ потребує постійної комунікації з різними людьми, з деякими вона може бути дуже непростою. Часто це викликає сильне розумове та емоційне перевантаження;
- робота під тиском термінів може призвести до стресу;
- високий рівень конкуренції попри високий попит на ринку праці;

- робота з високим рівнем невизначеності, що стає неабияким викликом для БА;
- високі вимоги до рівня англійської, що є радше бар'єром для входу в професію, ніж недоліком.

Які софт та хард скіли повинен мати бізнес-аналітик

Вимоги до посади ВА можуть відрізнятися від компанії до компанії. Зазвичай hard навички бізнес аналітика в ІТ включають (але не обмежуються) такими знаннями:

1. Розуміння бізнес-процесів: ідентифікація та аналіз ключових процесів, визначення слабких місць та пропозицій щодо покращення, вміння користуватися інструментами моделювання бізнес-процесів (BPMN, UML).
2. Requirements та changes management: виявлення вимог, опис, затвердження, атрибути якості, управління змінами, ризиками протягом проекту.
3. Вміння формувати скоуп проекту, щоб забезпечити відповідність встановленим цілям, уникати перевитрат.
4. Грамотне складання документації (BRD, SRS, User Stories, Use Cases тощо), яка допомагає розробникам та іншим членам ІТ-команди точно розуміти, що потрібно зробити і які бізнес-цілі мають бути досягнуті.
5. Вміння працювати з інструментами аналітики та візуалізації даних (Excel, SQL, Power BI, Tableau тощо). Аналітик повинен вміти витягувати з даних корисну інформацію, що допомагає приймати обґрунтовані бізнес-рішення.
6. Вміння користуватися інструментами для створення презентацій (Google Slides, Powerpoint тощо)
7. Розуміння Business Domain, вміння швидко вчитися, щоб розбиратися у різних доменах.
8. Розуміння SDLC та знання методологій розробки (Agile, Scrum, Kanban): вони допомагають БА ефективно інтегруватися у процес розробки та співпрацювати з розробниками та іншими стейкхолдерами.
9. Розуміння технічних аспектів розробки. Хоча робота бізнес-аналітика належить до нетехнічних спеціальностей у сфері ІТ, важ-

ливо мати розуміння технологій, що використовуються в проєктах. Базові знання основ програмування, розуміння відмінностей між різними платформами (web, mobile, desktop), основ тестування, API тощо дозволять аналітику більш ефективно взаємодіяти з технічною командою і вносити пропозиції, що враховують технічні реалії.

10. Вміння користуватися популярними інструментами в роботі будь-якої IT-команди (Slack, Jira, Confluence тощо).

11. Розуміння специфічної термінології в IT сфері, щоб “розмовляти однією мовою” з колегами.

12. Високий рівень англійської мови (від Intermediate+), адже бізнес-аналітик в IT часто спілкується з іноземними замовниками.

13. Бажано — знання UI/UX-дизайну та проєктного менеджменту.

Не менш важливими (а інколи навіть і важливішими) за hard skills є софт скіли:

1. Навички ефективної комунікації із командою розробки, замовниками та іншими учасниками проєкту.

2. Здатність бути командним гравцем.

3. Лідерські якості, вміння делегувати завдання та контролювати їх виконання.

4. Навички активного слухання.

5. Емпатія.

6. Вміння коректно, аргументовано та чітко висловлювати думку та надавати зворотний зв'язок.

7. Логічне та критичне мислення.

8. Сильні аналітичні здібності.

9. Вміння адекватно реагувати на критику.

10. Управління конфліктами та розв'язання проблем.

11. Вміння адаптуватися у швидкозмінному IT-середовищі.

12. Навички ефективної презентації.

13. Проактивність.

14. Саморозвиток.

15. Уважність до деталей.

16. Відповідальність.

17. Самоорганізація, тайм-менеджмент.

18. Вміння знаходити креативні рішення, пропонувати інновації та покращення.

Завдання до лабораторних робіт

Лабораторна робота №1. Визначення вимог ІС. Аналітичне обґрунтування вимог.

Завдання. За вказаною темою проекту визначити вимоги для формування інформаційної системи. Провести аналітичне дослідження сформульованих вимог та аналіз існуючого програмного забезпечення того ж призначення.

Лабораторна робота №2. Визначення типу ІС відповідно до вимог і потреб користувача. Обґрунтування обраного типу.

Завдання. Обґрунтувати аналітично обраний тип Інформаційної системи, визначити пріоритети вимог користувачів.

Лабораторна робота №3. Проектування ІС, планування ІС, організацію контролю виконання розробки ІС.

Завдання. Скласти план виконання розробки ІС
Оцінка ризиків виконання розробки ІС.

Список використаної літератури

1. 13 Common Project Risks and How to Tackle Them | Simplilearn. [Електронний ресурс]. – Режим доступу: <https://www.simplilearn.com/common-project-risks-article>
2. 13 Common Project Risks and How to Tackle Them | Simplilearn. [Електронний ресурс]. – Режим доступу: <https://www.simplilearn.com/common-project-risks-article>
3. Project Management Evolution Research Directions. (б. д.). [Електронний ресурс]. – Режим доступу: <https://www.pmi.org/learning/library/project-management-evolution-research-directions-8348>
4. The Ultimate Guide To SWOT Analysis For Business And Why It Matters. [Електронний ресурс]. – Режим доступу: <https://www.simplilearn.com/swot-analysis-for-business-article>
5. What is Application Software? (With Examples) | Simplilearn. [Електронний ресурс]. – Режим доступу: <https://www.simplilearn.com/tutorials/programming-tutorial/what-is-application-software>
6. What Is Kanban? An Overview Of The Kanban Method. [Електронний ресурс]. – Режим доступу: <https://www.nimblework.com/kanban/what-is-kanban/>
7. What is Risk in Project Management? (б. д.). [Електронний ресурс]. – Режим доступу: <https://www.wrike.com/project-management-guide/faq/what-is-risk-in-project-management/>
8. What is Risk Management and Why is it Important? (б. д.). Security. [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/searchsecurity/definition/What-is-risk-management-and-why-is-it-important>
9. Zulqadar, A. (2018, Серпень 10). SDLC Waterfall Model: The 6 phases you need to know about. Rezaid. [Електронний ресурс]. – Режим доступу: <https://rezaid.co.uk/sdlc-waterfall-model/>
10. Project Management Institute, A Guide to the Project Management Body of Knowledge, Fifth Edition, 2013, p. 331-332

Навчальне видання

Укладач:

Піддубна Лариса Андріївна

Інформаційні системи та технології у системному аналізі

**Конспект лекцій та завдання
до лабораторних робіт**

Відповідальний за випуск **Черевко І.М.**
Комп'ютерний набір **Піддубна Л.А.**