

*Крижановський Є. М.
Ящолт А. Р.
Жуков С. О.
Козачко О.М.*



Моделювання бізнес-процесів та управління ІТ-проектами

Міністерство освіти і науки України
Вінницький національний технічний університет

**Є. М. Крижановський, А. Р. Яшолт,
С. О. Жуков, О. М. Козачко**

**МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ ТА
УПРАВЛІННЯ ІТ-ПРОЕКТАМИ**

Електронний навчальний посібник

Вінниця
ВНТУ
2018

УДК 338.28:303.447.3(075)

ББК 32.97 : 22.16+26.22

К 74

Рекомендовано Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 10 від 26.08.2018 р.) до використання у навчальному процесі.

Рецензенти: **Бісікало О. В.**, доктор технічних наук, професор
Козловський С. І., доктор економічних наук, професор
Варчук І. В., кандидат технічних наук, доцент

Крижановський, Є.М.

К 74 Моделювання бізнес-процесів та управління ІТ-проектами : навчальний посібник [Електронний ресурс] / Є. М. Крижановський, А.Р. Яцолт, С.О. Жуков, О. М. Козачко – Вінниця : ВНТУ, 2018. – (PDF, 91 с.)

У навчальному посібнику викладено основні теоретичні відомості про методи та засоби моделювання бізнес-процесів та управління ІТ-проектами, а також практичні рекомендації для застосування сучасних технологій при моделювання бізнес-процесів та управлінні ІТ-проектами різної складності. Наведено перелік контрольних запитань для перевірки набутих теоретичних знань та практичних навичок.

Посібник рекомендується для студентів, які навчаються за спеціальностями 126– «Інформаційні системи та технології», 124 - «Системний аналіз», наступних дисциплін: Організація та управління ІТ проектами, Автоматизація бізнес-процесів, Управління ІТ-проектами, Моделювання бізнес-процесів та управління ІТ-проектами.

УДК 338.28:303.447.3(075)

ББК 32.97 : 22.16+26.22

Навчальне самостійне електронне мережне видання

Крижановський Євгеній Миколайович

Яцолт Андрій Русланович

Козачко Олексій Миколайович

Жуков Сергій Олександрович

Моделювання бізнес-процесів та управління ІТ-проектами

Навчальний посібник

Оригінал-макет підготовлено А. Р. Яцолтом

Електронний ресурс PDF.

Підписано до видання 25.07.2018 р. Зам. № P2018-024

Видавець та виготовлювач - Вінницький національний технічний університет,

Інформаційний редакційно-видавничий центр. ВНТУ, ГНК, к.114,

Хмельницьке шосе, 95, м. Вінниця, 21021,

тел. (0432) 65-18-06.

press.vntu.edu.ua;

Email: irvc.vntu@gmail.com.

Свідоцтво суб'єкта видавничої справи серія ДК № 3516 від 01.07.2009 р.

© Є. М. Крижановський, А. Р. Яцолт, С.О. Жуков, О. М. Козачко, 2018

ЗМІСТ

ВСТУП.....	4
1 ПРИНЦИПИ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ.....	5
1.1 Основні положення управління проектами.....	5
1.2 Класифікація ІТ-проектів, їх особливості.....	7
1.3 Життєвий цикл проекту.....	10
1.4 Учасники проекту.....	12
1.5 Форми організаційної структури.....	15
1.6 Стандарти управління проектами.....	17
Контрольні питання?.....	21
2 ПРАКТИЧНІ АСПЕКТИ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ.....	22
2.1 Процеси управління проектами.....	22
2.2 Моделі управління проектами.....	27
2.3 Документація проекту.....	36
2.4 Управління ризиками проекту.....	39
2.5 Програмні засоби для управління проектами.....	47
Контрольні питання?.....	50
3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ.....	51
3.1 Функціональне моделювання SADT (IDEF0).....	51
3.2 Моделювання процесів (IDEF3).....	64
3.3 Моделювання потоків даних (DFD).....	71
3.4 ARIS.....	78
3.5 Метод технології Rational Unified Process (UML).....	80
3.6 Метод Ericsson-Penke.....	82
3.7 Основні принципи та розвиток моделювання бізнес-процесів.....	83
Контрольні питання?.....	85
Глосарій.....	86
Література.....	88

Вступ

Моделювання бізнес-процесів та управління ІТ-проектами є одними з найпоширеніших напрямків застосування сучасних технічних засобів в галузі інформаційних технологій.

Моделювання бізнес-процесів в умовах сучасності неможливі без використання комп'ютерної техніки та сучасних програмних засобів. Сучасні пакети прикладних програм дозволяють здійснювати обробку даних з використанням традиційних та сучасних математичних методів для Моделювання бізнес-процесів. Застосування сучасних інформаційних технологій також є необхідним для здійснення ефективного системного управління ІТ-проектами.

Посібник орієнтовано на студентів комп'ютерних спеціальностей, які повинні вміти, використовуючи сучасні програмні середовища для моделювання бізнес-процесів різної складності, а також для здійснення управління ІТ-проектами.

Мета даного посібника – ознайомити студентів із основними знаннями та навичками, необхідними для здійснення моделювання бізнес-процесів різної складності різної складності, а також для здійснення управління ІТ-проектами..

Матеріал посібника може бути корисним і для слухачів другої вищої освіти інформаційних систем та технологій, чи системного аналізу та студентам інших спеціальностей, які недостатньо володіють основними знаннями та навичками роботи з найбільш поширеними в Україні програмними засобами, які можуть бути використані для проведення моделювання бізнес-процесів та управління ІТ-проектами. Також буде корисним слухачам наступних курсів: Організація та управління ІТ-проектами, Автоматизація бізнес-процесів, Управління ІТ-проектами, Моделювання бізнес-процесів та управління ІТ-проектами.

1 Принципи управління ІТ-проектами.

1.1 Основні положення управління проектами

Проекти часто використовуються як засіб прямого або непрямого або декількох з наступних стратегічних міркувань [1]:

- вимога ринку (наприклад, автомобілебудівна компанія авторизує проект по виготовленню більш економічних автомобілів у відповідь на брак бензину);
- стратегічна можливість / бізнес-потреба (наприклад, тренінгова компанія авторизує проект зі створення нового курсу навчання з метою збільшення доходів);
- соціальна потреба (наприклад, неурядова організація в країні, що розвивається, авторизує проект з надання систем питного водопостачання, туалетів і санітарної освіти співтовариствам, що страждають від високого рівня інфекційних захворювань);
- захист навколишнього середовища (наприклад, державна компанія авторизує проект по створенню нового сервісного центру для електромобілів, які сприяють скороченню забруднення навколишнього середовища);
- вимога замовника (наприклад, компанія-виробник електроенергії для громадського користування авторизує проект з будівництва нової підстанції для електропостачання нового промислового району);
- технологічний прогрес (наприклад, виробник комп'ютерної техніки авторизує проект по розробці більш швидкодіючого, економічного і компактного ноутбука з використанням досягнень в технології виготовлення комп'ютерної пам'яті і електронних компонентів);
- юридична вимога (наприклад, виробник хімічних речовин авторизує проект по розробці керівних вказівок щодо поводження з новим токсичним матеріалом).

Організації здійснюють керівництво для визначення стратегічного напрямку і параметрів продуктивності. Даний стратегічний напрям надає мету, очікування, завдання і дії, необхідні для керівництва діяльністю організації, і приводиться у відповідність з бізнес-цілями. Роботи з управління проектом повинні бути приведені у відповідність з напрямком організації на верхньому рівні, і в разі його зміни мети проекту повинні бути переглянуті. В умовах виконання проекту зміни з метою проекту впливають на ефективність і успіх проекту.

Основними принципами нової концепції управління є [1]:

- 1) поглиблення рівня обґрунтованості прийнятих інвестиційних рішень, використовуючи механізм різноманітних і багатофакторних (технологічних, економічних, соціальних, екологічних та інших) оцінок;
- 2) високий ступінь координації та контролю робіт в процесі виконання проекту;

3) систематичний аналіз і врахування зовнішніх змін (кон'юнктури ринку по всіх видах ресурсів, непередбачених обставин і негативних факторів) при реалізації проектів.

Управління проектами базується на системному підході, що дає можливість декомпозиції і структуризації проекту будь-якої складності при прийнятті рішень в складних умовах (ситуаціях). Особливість методології управління проектами полягає в зосередженні прав і відповідальності за досягнення цілей проекту на одній людині або невеликій групі. Ці права і обов'язки здійснює керівник проекту. При цьому забезпечується [1]:

1. Визначення всіх видів робіт, необхідних для досягнення цілей проекту, їх структуризація і визначення взаємозв'язків.

2. Складання і контроль кошторису витрат по реалізації проекту.

3. Розробка та контроль графіків робіт, необхідних для досягнення бажаного результату.

4. Розподіл ресурсів, виділених для реалізації проекту, в умовах невизначеностей і ризиків.

5. Управління якістю виконання всіх робіт по проекту, включаючи досягнення планованої якості продукції, пропонованої проектом, і виконання інших вимог замовника.

6. Управління ризиком (ризиками) на всіх етапах здійснення проекту.

7. Забезпечення зв'язку з клієнтами, замовниками, споживачами продукції, з різними групами і особами, залученими до проекту (соціальні групи, місцеве населення, влади, засоби масової інформації) для вирішення всіх питань, пов'язаних з досягненням успіху проекту.

При розгляді та вивченні діяльності з управління реалізацією проектів можна виділити ряд підходів, що визначають структуризацію цієї діяльності. Найбільш поширені з них: функціональний, динамічний і предметний підходи.

а) Функціональний підхід відображає загальний підхід до проблеми управління і передбачає розгляд основних управлінських функцій або видів управлінської діяльності при здійсненні проектів у відповідності з класифікацією, прийнятою в менеджменті. У той же час, специфіка управління проектами проявляється в певній деталізації окремих функцій менеджменту. З урахуванням цього в рамках функціонального підходу розглядаються управлінські дії, які визначаються як функції управління проектами, включаючи: аналіз, планування, організацію, контроль і регулювання [1].

б) Предметний підхід визначає структуризацію управлінського процесу по об'єктах проекту, на які направлено управління. В рамках предметного підходу розглядаються два типи об'єктів: виробничі об'єкти (перший тип) і елементи, пов'язані з діяльністю щодо забезпечення реалізації проекту (другий тип). Об'єктами другого типу можуть бути: фінанси, кадри, маркетинг, контроль, ризики, матеріальні ресурси, якість, інформація та інші елементи, що забезпечують отримання бажаного результату при реалізації проекту. Основні об'єкти першого і другого типів розглядаються в міру необхідності при вивченні дисципліни.

с) Динамічний підхід (відомий в літературі як проектний менеджмент – project management), передбачає розгляд у часі процесів, пов'язаних з реалізацією основних стадій і етапів реалізації проекту: аналіз проблеми, розробка концепції проекту, проектування, будівництво, монтаж, налагоджувальні роботи, експлуатація та завершення проекту. Деталізація розглянутих процесів стосовно до конкретного проекту (етапів і стадій робіт здійснення проекту) може змінюватися в широких межах [1].

1.2 Класифікація ІТ-проектів, їх особливості

Для зручності аналізу і синтезу проектів, а також систем управління ними безліч різноманітних ІТ-проектів класифікується за різних підстав. У науковій літературі зустрічаються різні підходи до класифікації проектів. Найбільш часто вживаною класифікацією є [2-12]:

1) Клас. За складом і структурі проектів та його предметної області проекти поділяються на монопроекти, мультипроекти, мегапроекти.

2) Тип. За основними сферами діяльності, в яких здійснюється проект, виділяють технічні, організаційні, економічні, соціальні та змішані проекти.

3) Вид. За характером предметної галузі проекти поділяються на: інвестиційні, науково-дослідні, навчально-освітні, змішані.

4) Масштаб. За розмірами самого проекту, кількістю учасників і ступеня впливу на навколишній світ проекти поділяють на: дрібні, середні, великі й дуже великі проекти.

5) Тривалість. За тривалістю періоду здійснення проекти поділяються на: короткострокові, середньострокові і довгострокові проекти.

6) Складність. За ступенем складності: прості, складні і дуже складні. Використання зазначеної класифікації не в повній мірі дає уявлення про відмінності ІТ-проектів від інших видів проектів. Як наслідок, виникає необхідність індивідуалізації представленої класифікаційної системи.

ІТ-проекти можна розділити на три основні типи [2, 6]:

1) розробка програмного забезпечення, комп'ютерних програм, процедур і, можливо, відповідної документації та даних, відносяться до функціонування інформаційної системи;

2) розробка продуктів орієнтованих на он-лайн використання;

3) розробка різних додатків, частіше яскраво вираженого комерційного спрямування.

Усі три види мають свою специфіку. Грунтуючись на такий поділ, можна виділити наступні класифікаційні ознаки ІТ-проектів [2-12]:

1) За характером змін: прості (проект, здійснення змін у структурі та змісті яких, що не призводить до зміни його вартісних і часових параметрів); середні (проект, здійснення змін у структурі та змісті яких, призводить до не суттєво зміни його вартісних і часових параметрів); складні (проект, здійснення змін у

структурі та змісті яких призводить до істотної зміни його вартісних і часових параметрів).

2) За масштабом: малі (вартістю до 10 тис. Грн.); середні (вартістю 10-50 тис. Грн.); великі (вартістю 50-100 тис. Грн.); значні (вартістю 100-1000 тис. Грн.); дуже значні (вартістю понад 1 млн. Грн.).

3) За тривалістю: Короткострокові - тривалістю до 1-го року; Середньострокові - тривалістю від 1-го до 3-х років; Довгострокові – тривалістю понад 3-х років.

4) За стадіями життєвого циклу системи: оформлення задуму і концепції; формулювання вимог до системи; розробка системи; введення системи в експлуатацію; підтримка існуючої системи.

5) По виду продукту: система; програмний продукт; технічні засоби; програмно-технічні комплекси; матеріали, роботи та послуги.

6) За функціональним призначенням (орієнтованість продуктів ІТ-проектів на напрямки діяльності замовника): виробничі; технологічні; фінансові; дослідні; маркетингові; з управління персоналом; з управління проектами; ігрові; комбіновані.

7) За глибиною взаємного проникнення бізнесу замовника і підрядника: ауторсинг; рішення «під ключ»; спільні проекти; сервісна модель; аудит і консалтинг.

8) По виду замовника: держсектор; медицина; освіту; дрібний бізнес; великий бізнес; логістика, сфера послуг; роздріб; енергетика; банківський сектор; транспорт і зв'язок; промисловість; оборонна промисловість і т.п.

9) По виду автоматизованих процесів: основні і допоміжні; технологічні та офісні; управлінські; аналітичні; транзакційні; реального часу; з тим чи іншим акцентом на обчислювальну обробку; передача даних; організація зберігання; обробка медіа-контенту; забезпечення безпеки і т.п.

10) За ступенем складності: монопроекти, мультипроекти і мегапроекти з притаманними їм загальноприйнятими характеристиками.

11) За територіальним поширенню: мононаціональною – продукт проекту орієнтований на територіальні регіони зі схожою ментальністю; полінаціональність – продукт проекту орієнтований на територіальні регіони з різною ментальністю.

12) За рівнем впливу розробки інтерфейсу на проект: Низький - результат розробки інтерфейсу має незначний вплив на оцінку проекту в цілому, займає невеликий відрізок часу в життєвому циклі проекту; Середній - результат розробки інтерфейсу має значний вплив на оцінку проекту в цілому, займає значний відрізок часу в життєвому циклі проекту; Високий - результат розробки інтерфейсу має критично важливе вплив на оцінку проекту в цілому, займає значний відрізок часу в життєвому циклі проекту, може вплинути на прийняття рішення про закриття проекту.

Управління проектами в області інформаційних технологій за останній час завоювало визнання як найкращий метод планування та управління реалізацією інвестиційних проектів. За американськими оцінками застосування методології

Управління Проектами забезпечує високу надійність досягнення цілей проекту і на 10-15% скорочує витрати на його реалізацію [1].

У світі накопичено величезний досвід застосування управління проектами в області інформаційних технологій. Зокрема, ця методологія застосовується у всіх великих ІТ-компаніях світу. Програмні засоби для управління проектами встановлені на мільйонах комп'ютерах в усьому світі – тільки пакет Microsoft Project встановлений більш ніж на два мільйони комп'ютерів. Асоціація управління проектами Project Management Institute (Інститут Управління Проектами) об'єднує близько 40 тисяч членів і має відділення майже на всіх континентах [1, 11].

Розглянемо основні поняття і методи управління проектами.

Проект – це тимчасове підприємство, призначене для створення унікальних продуктів або послуг.

В даному контексті "Тимчасове" означає, що у кожного проекту є початок і неодмінно настає завершення, коли досягаються поставлені цілі, або приходиться розуміння, що ці цілі не можуть бути досягнуті.

В даному контексті "Унікальних" означає, що створювані продукти або послуги відрізняються від інших аналогічних продуктів і послуг. Приклади проектів: розробка нового обладнання, розробка або впровадження програмних засобів і т.д.

Управління проектами – це застосування знань, досвіду, методів і засобів до робіт проекту для задоволення вимог, що пред'являються до проекту, і очікувань учасників проекту. Щоб задовольнити ці вимоги і очікування необхідно знайти оптимальне поєднання між усіма характеристиками проекту [11].

Управління проектами підпорядковується чіткій логіці, яка зв'язує між собою різні області знань і процеси управління проектами.

Кожен проект приводить до створення унікального продукту, послуги або результату.

Перш за все, у проекті обов'язково є одна або кілька цілей. Під цілями розуміється не тільки кінцеві результати проекту, а й обрані шляхи досягнення цих результатів (наприклад, застосовувані в проекті технології, системи управління проектом) [1].

Досягнення цілей проекту може бути реалізовано різними способами. Для порівняння цих способів необхідні критерії успішності досягнення поставлених цілей. Зазвичай в число основних критеріїв оцінки різних варіантів виконання проекту входять терміни і вартість досягнення результатів. При цьому заплановані цілі і якість зазвичай служать основними обмеженнями при розгляді та оцінці різних варіантів. Звичайно, можливе використання і інших критеріїв, і обмежень – зокрема, ресурсних [1].

Застосування технологій і ресурси проекту можна віднести до основних важелів управління проектами. Крім цих основних існують і допоміжні засоби, призначені для управління основними. До таких допоміжних важелів управління можна віднести, наприклад, контракти, які дозволяють залучити потрібні ресурси в потрібні терміни. Крім того, для управління ресурсами необхідно за-

безпечити ефективну організацію робіт. Це стосується структури управління проектом, організації інформаційної взаємодії учасників проекту, управління персоналом.

Інформація, яка використовується в управлінні проектами, зазвичай не буває стовідсотково достовірною. Облік невизначеності вихідної інформації необхідний і при плануванні проекту і для грамотного укладання контрактів. Аналізу та обліку невизначеностей присвячений аналіз ризиків [1, 11].

Управління проектом здійснюється за допомогою логічно згрупованих процесів управління проектом, об'єднаних в 5 груп [1]:

- Процеси ініціації – ухвалення рішення про початок виконання проекту;
- Процеси планування – визначення цілей і критеріїв успіху проекту і розробка робочих схем їх досягнення;
- Процеси виконання – координація людей і інших ресурсів для виконання плану;
- Процеси моніторингу і контролю – визначення відповідності плану і виконання проекту поставленим цілям і критеріям успіху і прийняття рішень про необхідність застосування коригувальних дій і визначення необхідних коригувальних впливів, їхнє узгодження, затвердження і застосування;
- Процеси закриття – формалізація виконання проекту і підведення його до впорядкованого фіналу.

Практично методологія управління проектами допомагає [1]:

- обґрунтувати доцільність інвестицій,
- розробити оптимальну схему фінансування робіт,
- скласти план робіт, що включає терміни виконання робіт, споживання ресурсів, необхідні витрати,
- оптимально організувати виконання робіт і взаємодію учасників проекту,
- здійснювати планування і управління якістю,
- здійснювати аналіз і управління проектними ризиками,
- оптимально планувати і управляти контрактами,
- аналізувати відхилення фактичного ходу виконання робіт від запланованого і прогнозувати наслідки виникнення відхилень,
- моделювати коригувальні дії на інформаційних моделях проектів і приймати обґрунтовані управлінські рішення,
- вести архіви проектів і аналізувати досвід їх реалізації, який може бути використаний в інших проектах, і т.д.

1.3 Життєвий цикл проекту

Життєвий цикл проекту – набір фаз, через які проходить проект з моменту його ініціації до моменту закриття. Фази, як правило, є послідовними, а їх назви та кількість визначаються потребами в управлінні і контролі організації або організацій, залучених до проекту, характером самого проекту і його прикладною областю. Проект може бути розбитий на фази в залежності від функці-

ональних або часткових цілей, проміжних результатів, визначених контрольних подій всередині загального змісту робіт або доступності фінансів. Фази, як правило, обмежені в часі, і мають початкову і кінцеву або контрольну точку. Життєвий цикл може документуватися в рамках методології. Життєвий цикл проекту може визначатися або формуватися унікальними аспектами організації, галузі або використовуваної технології. У той час як кожен проект має певний початок і закінчення, конкретні результати і дії, що мають місце в цьому проміжку, широко варіюються для кожного проекту. Життєвий цикл забезпечує базову структуру для управління проектом, незалежно від включених в нього конкретних робіт [1].

Проекти розрізняються за розміром і складністю. Всі проекти можуть мати наступну структуру життєвого циклу (рис. 1.1):

- початок проекту;
- організація та підготовка;
- виконання робіт проекту;
- завершення проекту.

Дана узагальнена структура життєвого циклу часто згадується при комунікаціях з вищим керівництвом або іншими сторонами, які менш обізнані про деталі проекту. Не слід плутати її з групами процесів управління проектом, тому що процеси в групі процесів складаються з дій, які можуть виконуватися і повторюватися в кожній фазі проекту, а також бути характерними для проекту в цілому. Життєвий цикл проекту не залежить від життєвого циклу продукту, виробленого або модифікованого в результаті виконання проекту. Однак проект повинен враховувати поточну фазу життєвого циклу продукту. Це високорівневе представлення забезпечує єдину систему відліку при порівнянні проектів, навіть якщо вони різноманітні за своєю природою [1].

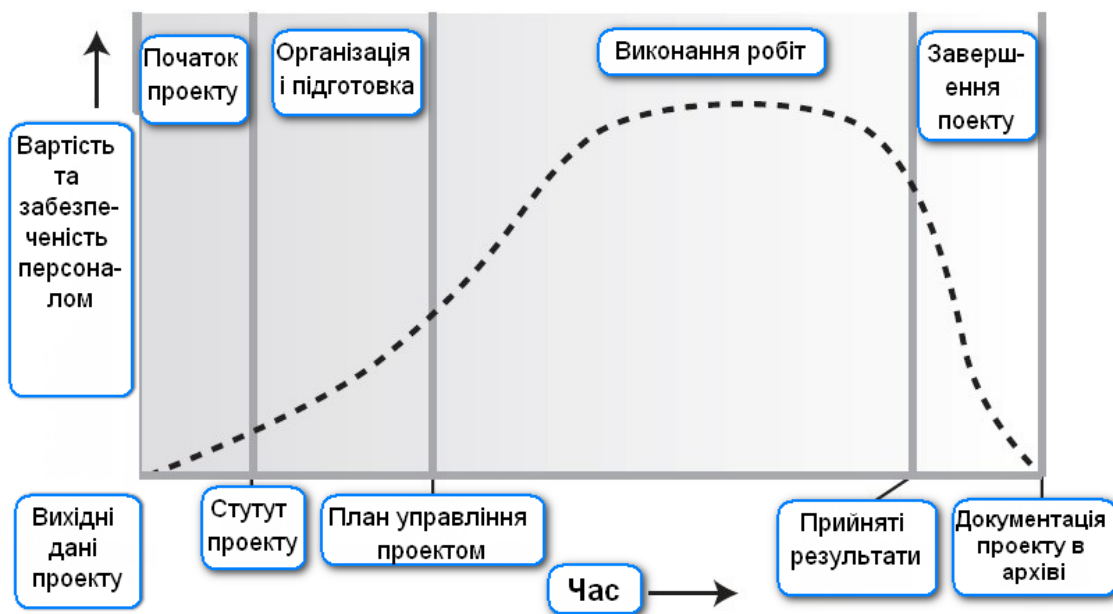


Рис. 1.1. Життєвий цикл проекту

Нині для підприємств характерні такі проблеми управління ІТ-проектами [13]:

- 1) кількість проектів, керованих ІТ-підрозділами, завелика, немає достатньої кількості кваліфікованих менеджерів;
- 2) наявна методологія не забезпечує відповідності ІТ-проектів цілям компанії, не створена основа для оцінки повернення інвестицій, вкладених в ІТ;
- 3) роботи виконуються за різними технологіями, рівень керованості недостатній;
- 4) відсутня оперативна аналітична звітність і накопичення досвіду за проектами.

Серед проблем, суттєвих з погляду успіху проекту в цілому, можна виділити такі: прагнення керівництва підприємства до досягнення бізнес-цілей, поставлених перед ІТ- проектом; відповідність функціональності ІС потребам бізнесу і керівництва підприємства; управління в межах проекту (терміни, бюджет, склад робіт), тому що ІТ-проекти мають властивість "розповзатися". Це відбувається з різних причин, серед яких можуть бути: зміна організаційної структури і реінжиніринг бізнес-процесів підприємства, відсутність деталізованого проекту ІС, неконтрольований потік вимог у процесі виконання проекту з боку замовника, зміна проектних рішень розробниками системи в ході її реалізації, відсутність типізації проектних рішень; взаємодія різних груп учасників ІТ-проекту. У великому ІТ-проекті може брати участь кілька сотень і тисяч осіб. Умовно їх можна розділити на декілька груп: розробники ІС, аналітики і методологи, функціональні фахівці замовника, фахівці з навчання і супроводу ІС. Кожна група є носієм певної категорії знань і виконує певні ролі у проекті. Внаслідок різного професійного досвіду, термінологічного базису, покладених функціональних завдань спілкування між представниками різних груп часто викликає проблеми і може призводити до конфліктних ситуацій; об'єктивний моніторинг поточного стану проекту. Часто керівництво ІТ-проекту не має ефективних засобів доступу до адекватної інформації про поточний стан проекту, що перешкоджає формуванню своєчасних управлінських рішень щодо проекту; ведення проектної документації. У багатьох ІТ-проектах пред'являються невисокі вимоги до ведення проектної документації. Це негативно позначається на етапах супроводу і розвитку ІС, навчанні кінцевих користувачів, призводить до зміни учасників проектної команди; склад і кваліфікація учасників проекту, функціональні обмеження типових рішень ІС тощо [13].

1.4 Учасники проекту

Головним відповідальним і головною дійовою особою в проекті, через мізки якого повинні пройти всі елементи проекту, є керівник (менеджер) проекту. Він призначається на першій фазі (рис. 1.1) одночасно з затвердженням Статуту проекту [1].

Роль керівника проекту – особи, відповідальної за успіх всього проекту набагато ширша ролі функціонального менеджера. Різниця визначається тим, що менеджер проекту [1]:

- управляє тимчасовою діяльністю і командою тимчасових учасників фахівців з різних галузей знань;
- має заздалегідь поставлену мету;
- обмежений термінами, бюджетом і технічними умовами (вимогами до продукту і його якості);
- управляє інтеграцією всіх елементів проекту, сам планує роботи і використання ресурсів і сам втілює їх у життя;
- керує створенням нового унікального продукту, властивості якого можуть уточнюватися в міру просування;
- не завжди розуміється в тонкощах реалізації продукту.

Для ефективного виконання своєї ролі менеджер проекту повинен володіти знаннями і реальним досвідом в наступних областях:

- проектний менеджмент – технологічні знання і методи управління проектами;
- загальний менеджмент – знання і методи повсякденного управління компаніями;
- конкретні прикладні, предметні області.

Проектний менеджмент (управління проектами) - організаційна діяльність з оперативного управління проектом в умовах обмежень.

Сьогодні проектний менеджмент багато в чому спирається на стандарти ANSI PMI PMBOK. Практичне володіння стандартами і переліком знань з управління проектами дає менеджеру проекту значну перевагу перед іншими менеджерами, в т.ч. при працевлаштуванні [14-15].

Загальний менеджмент охоплює всі аспекти управління повсякденною діяльністю підприємства. Приблизний перелік знань і навичок, що входять в програми MBA (Master of Business Administration) – Майстер ділового адміністрування, демонструє ємність завдань загального менеджменту [1]:

- планування часу і делегування повноважень;
- особиста ефективність, проведення нарад і особиста мотивація;
- подолання проблем і прийняття рішень;
- мистецтво ведення переговорів, ефективного спілкування, публічних виступів, листування;
- фінансовий аналіз і бухгалтерський облік, оцінка інвестицій;
- бюджетування, бюджетний контроль;
- управління продажами і маркетинг;
- управління запасами і незавершеним виробництвом;
- дослідження і розробки, фінансовий аналіз проектів;
- фінансування, управління прибутком;
- управління персоналом, організаційне управління;
- конкурентна ринкова стратегія, оцінка бізнесу, покупка і продаж компаній;

- стратегічне планування бізнесу, розробка бізнес-планів;
- методики управління підприємством, в т.ч. методика BSC (Balanced Score Card) - СЗП (Система збалансованих показників);
- управління змінами, кризовий менеджмент та ін.

Знання та навички загального менеджменту складають хорошу основу для оволодіння знаннями і навичками управління проектами. Проектний менеджмент розвинувся і розглядається як окрема галузь знань, навичок, компетенцій, що стоїть поруч і на службі традиційних областей професійної діяльності, таких як будівництво і архітектура, ІТ-індустрія, соціологія і т.д [1].

Крім менеджера проекту, в проекті зазвичай беруть участь безліч інших осіб і організацій. До учасників проекту відносяться фізичні та юридичні особи, залучені до проекту, а також особи, які мають вплив (позитивне і негативне) на проект і його результати. Їх в термінології PMI PMBOK називають стейкхолдерами проекту [15].

Виявити всіх стейкхолдерів проекту та інтереси часто буває важко і це є однією з перших завдань менеджера проекту. До ключових стейкхолдерів відносяться [1]:

- менеджер проекту – особа, відповідальна за кінцеві результати проекту та управляє проектом;
- замовник – фізична або юридична особа – майбутній споживач продукту проекту;
- підрядник – юридична особа, співробітники якої виконують роботи проекту; виконуючою організацією може виступати як зовнішня організація, так і тимчасова структура всередині самої організації-замовника;
- спонсор – особа або група осіб (фізичних або юридичних), що забезпечує проект фінансовими та іншими ресурсами;
- члени команди проекту – група, яка виконує роботи проекту.

До завдань менеджера проекту входить і управління стейкхолдерами. Воно розглядається як профілактичне завдання, спрямоване на максимальне врахування інтересів стейкхолдерів, використання їх активності для досягнення цілей проекту, нейтралізації їх негативного впливу. Вона складається як мінімум з таких пунктів [1]:

- визначення стейкхолдерів, а також оцінка їх компетентності, знань і навичок;
- аналіз проекту на відповідність вимогам стейкхолдерів;
- залучення стейкхолдерів в проект різними шляхами: в якості експертів, в якості членів комісій щодо змін та ін.;
- якщо існують розбіжності між стейкхолдерами, то за допомогою компромісів проблема повинна вирішуватися на користь замовника.

1.5 Форми організаційної структури

Структура виконуючої організації накладає обмеження на процеси управління проектом і розподіл ресурсів. Позначимо дві протипожежні форми організаційної структури – функціональна (див. рис. 1.2) і проектна (див. рис. 3). На рисунках сірим кольором виділено персонал, який бере участь в проектах. У функціональній структурі кожен такий працівник має одного керівника і взаємодіє тільки з ним. Зазвичай функціональний підрозділ виконує частину робіт проекту або весь проект. Функціональний керівник вирішує всі питання розподілу ресурсів, завдань, взаємодії і залучення фахівців з інших функціональних підрозділів. Роль менеджера проекту розмита. У таких випадках роль координатора проекту покладається на експедитора проекту (або експедитора проекту). У той же час експедитор не може приймати самостійні рішення по проекту. Якщо ж призначається менеджер проекту, то він повинен мати певні повноваження управління командою, максимально чітко встановлені [1].

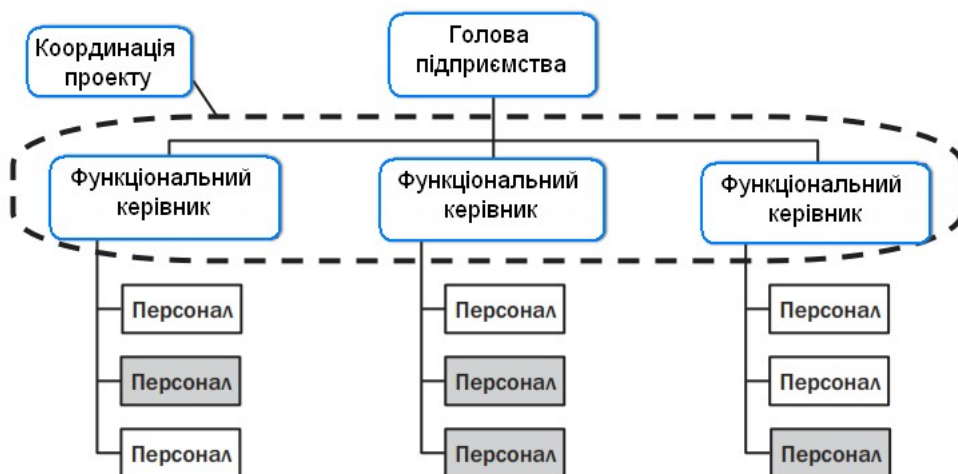


Рис. 1.2. Функціональна структура

У проектній структурі (див. рис. 1.3) весь персонал на проект зібраний в одному місці (команда проекту). Створені всі умови для тісної взаємодії персоналу різних спеціальностей. Менеджер проекту в значній мірі незалежний і володіє максимальними повноваженнями. У багатьох випадках таке формування організується у вигляді проектного офісу.

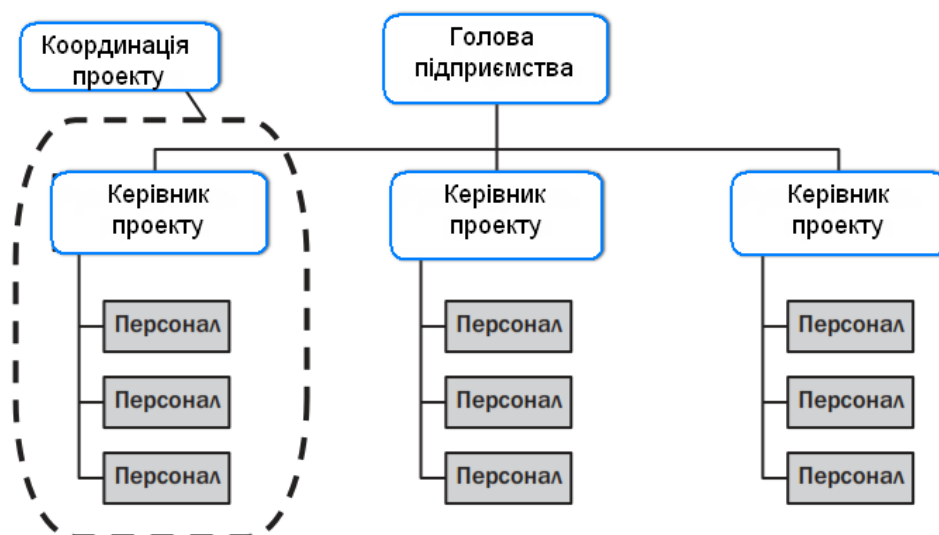


Рис. 1.3. Проектна структура

Функціональна структура більше орієнтована на виконання повторюваних циклічних робіт. Переваги функціональної структури стають недоліками проектною та навпаки – наведені нижче:

Таблиця 1.1. Порівняння організаційних структур

Проектна організаційна структура	Функціональна організаційна структура
Ефективна інтеграція, організація і контроль проекту	Складніший моніторинг і контроль проекту. Акцент на функціональну спеціалізацію зі шкодою для інтеграції та інших робіт проекту
Ефективна комунікація та рішення конфліктів	Неефективні комунікації по ієрархічній драбині
Націленість на проект і лояльність проекту	Розмитість пріоритетів виконання робіт
Імовірність браку професіоналізму по різних дисциплінах	Високий професіоналізм, але акцент на функціональну спеціалізацію зі шкодою для інтеграції та інших робіт проекту
Імовірність надлишку і менш ефективного використання ресурсів	Повна завантаженість фахівців і більш легке управління ними
Тимчасова робота	Наявність постійної роботи по завершення проекту
Одна відповідальна особа – менеджер проекту	Розмита відповідальність за результати проекту

Тому на практиці часто застосовують комбінацію зазначених вищеструктур – комбінована організаційна структура – або інші поєднання – слабку матричну структуру або сильну матричну структуру. Комбінована структура дозволяє мінімізувати недоліки описаних структур показана на рис. 1.4.

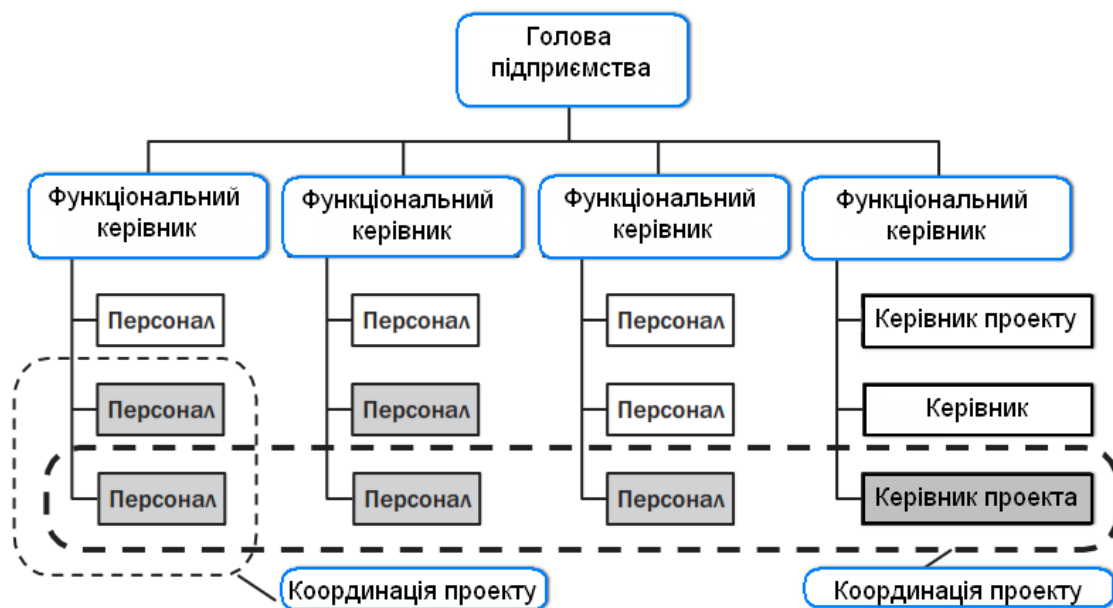


Рис. 1.4. Комбінована організаційна структура

У комбінованій структурі персонал на проект може мати подвійне підпорядкування: в рамках організації підпорядкування функціональному керівнику (менеджеру), а в рамках проекту – керівнику (менеджеру) проекту. Тому повинні бути чітко розмежовані повноваження між керівником проектів і функціональними керівниками. На рис. 4 керівник проекту чітко планує і погоджує участь персоналу в проекті з функціональними керівниками [1].

Якщо рівень організаційної культури організації не дозволяє розмежувати повноваження менеджерів проекту і функціональних менеджерів, не допускає нововведень для досягнення успіху, то кращим рішенням є створення незалежних проектних команд для виконання критично важливих проектів.

1.6 Стандарти управління проектами

Методологія управління проектами відбивається в стандартах управління проектами. В даний час існують наступні види стандартів [1]:

- міжнародні – стандарти, які отримали міжнародне значення в процесі свого розвитку або призначені для міжнародного використання;
- національні – створені для застосування всередині однієї країни або отримали загальнонаціональний статус в процесі свого розвитку;
- громадські – підготовлені і прийняті спільнотою фахівців;
- приватні – комплекси знань, пропаговані для вільного використання приватними особами, компаніями або установами;
- корпоративні – розроблені для застосування в межах однієї компанії або всередині групи споріднених компаній.

Міжнародні стандарти – найбільш повні системи, що включають, крім опису вимог до управління проектами, навчання, тестування, аудит, консалтинг та

інші елементи. Всеохоплюючих міжнародних стандартів управління проектами поки не існує, але найбільш відомі такі нормативні документи [15].

1. **Project Management Body of Knowledge (PMBOK)** Американського інституту управління проектами (Project Management Institute - PMI). Цей стандарт оновлюється приблизно один раз в чотири роки. Одна з найбільш поширених редакцій датується 2000 р., а найактуальніша, четверта, версія стандарту – The Guide to the PMBOK, 4th Edition – вийшла в кінці 2008 р. Стандарт був спочатку прийнятий Американським національним інститутом стандартів (ANSI) в якості національного стандарту в США, а в даний час знайшов світове визнання [1, 15].

В основі стандарту лежить процесний підхід до управління проектами. Загальна безліч можливих процесів представляється у вигляді тривимірного простору. По осях координат відкладаються ті вимірювання, які згадуються в рамках стандартів. Можуть бути запропоновані й інші, наприклад, рівні управління, календарні періоди, тощо. Кожна точка цього простору є елементарний процес управління. Наприклад, "планування ризиків на стадії впровадження системи".

Вибрані елементарні процеси утворюють процедури управління проектами, які можуть бути побудовані за "осьовим" принципом.

Стандарт містить узагальнені принципи та підходи, які використовуються в області проектного менеджменту, формалізовані і структуровані таким чином, щоб їх можна було використовувати в більшості проектів в більшості випадків. Детально описуються дев'ять областей знань, пов'язаних з управлінням проектами [1]:

- управління інтеграцією проекту (Project Integration Management);
- управління змістом проекту (Project Scope Management);
- управління термінами проекту (Project time Management);
- управління вартістю проекту (Project Cost Management);
- управління якістю проекту (Project Quality Management);
- управління людськими ресурсами проекту (Project Human Resource Management);
- управління взаємодією в проекті (Project Communications Management);
- управління ризиками проекту (Project Risk Management);
- управління контрактами проекту (Project Procurement Management).

Кожна область знання включає в себе окремі процеси, що виконуються менеджером при реалізації проекту на тому чи іншому етапі. Процесно орієнтований підхід в управлінні проектами, використовуваний в стандарті і передбачає чіткий, формальний опис вхідних документів і даних, необхідних менеджеру для реалізації процесу, методів і засобів, які він може використовувати при його реалізації, і переліку вихідних документів процесу.

2. **IPMA Competence Baseline (ICB)** є міжнародним нормативним документом, що визначає систему міжнародних вимог до компетентності менеджерів проектів. Цей стандарт розроблений міжнародною асоціацією IPMA

(International Project Managers Association). На його основі проводиться розробка національних систем вимог до компетентності фахівців в країнах, які є членами IPMA. Національні системи вимог повинні відповідати ІСВ IPMA і офіційно затверджуватися (ратифікуватися) відповідними уповноваженими органами IPMA. Для 32 країн - членів IPMA він є основою для розробки національних збірок знань [1, 15].

ІСВ, на відміну від PMBOK, дотримується компетентнісного, діяльнісного підходу, тобто визначає області кваліфікації та компетентності в управлінні проектами, а також принципи оцінки кандидата на отримання сертифіката. ІСВ містить 42 елемента (28 основних і 14 додаткових), що визначають області вимог до знань, майстерності та професійного досвіду в менеджменті проектів.

ІСВ видано англійською, німецькою та французькою мовами. Основою для нього послужило кілька національних розробок: Body of Knowledge of APM (Великобританія); Beurteilungsstruktur, VZPM (Швейцарія); PM-Kanon, PM-ZERT / GPM (Німеччина); Criteres d'analyse, AFITEP (Франція).

3. Стандарт ISO 10006. Звернення до питань ефективності проектного управління об'єктивно виявило гостру потребу в розробці системи управління якістю проекту. При цьому особливого значення поряд з вимогами до якості кінцевого продукту стало надаватися якості процесів проекту, відсутність належної уваги до яких, призводило до не менш значущих негативних наслідків безпосередньо для створюваного продукту [1].

Стандарт ISO 10006 є основоположним документом із серії стандартів розглянутого профілю, підготовленим технічним комітетом ISO / TC 176 "Управління якістю і забезпечення якості" Всесвітньої федерації національних органів стандартизації (члени ISO).

Основний акцент зроблений на принцип ефективності проектування оптимального процесу і контролю цього процесу, а не на контролі кінцевого результату.

У цій серії стандартів процеси згруповані у дві категорії. До першої категорії віднесені процеси, пов'язані із забезпеченням продукту проекту (проектування, виробництво, перевірка). Опису останніх присвячений стандарт ISO 9004-1. Друга категорія охоплює безпосередньо процеси управління проектом і представлена стандартом ISO 10006.

Даний стандарт охоплює десять груп процесів управління проектом. Перша група являє процес розробки стратегії, який фокусує проект на задоволення потреб замовника і визначає напрямок ходу робіт. Друга група охоплює управління взаємозв'язками процесів. Решта вісім груп – це процеси, пов'язані з проектним завданням, термінами, витратами, ресурсами, кадрами, інформаційними потоками, ризиком і матеріально-технічним постачанням (закупівлями).

Міжнародний стандарт ISO 10006 орієнтований на проекти найширшого спектра – малі і великі, короткострокові і довгострокові, для різних навколишніх умов. Він не залежить від типу проектуваного продукту (включаючи технічні засоби, програмне забезпечення, напівфабрикати, послуги або їх поєднання). Це означає, що закладені в ньому рамкові вимоги вимагають подальшої

адаптації даного керівництва до конкретних умов розробки і реалізації окремого проекту [1].

Стандарт запозичує ключові визначення з ISO 8402, включаючи такі терміни, як проект, продукт проекту, план проекту, учасник проекту, процес, оцінка ходу робіт. Для всіх процесів управління проектом (планування, організація, моніторинг і контроль) застосовуються процеси і завдання менеджменту якості [15].

4. Стандарти зрілості управління проектами, теж набувають функції міжнародних. У 2004 р. PMI був випущений стандарт оцінки рівня зрілості організації з управління проектами ОРМЗ (Organization Project Management Maturity Model), що містить методологію визначення стану управління проектами в організації.

Термін «організаційна зрілість з управління проектами» описує здатність організації відбирати проекти та управляти ними таким чином, щоб це максимально ефективно підтримувало досягнення стратегічних цілей компанії.

ОРМЗ - це стандарт, який представляє собою всебічний підхід, який допомагає організаціям оцінювати і розвивати свої можливості щодо ефективної реалізації проектів. Він є свого роду ключем до організаційної зрілості управління проектами і містить три взаємозалежних елементи [1]:

- елемент «знання» (knowledge) являє собою сотні кращих практик з управління проектами, що характеризують ті чи інші рівні організаційної зрілості управління проектами;

- елемент «оцінка» (assessment) є інструментом, що допомагає організаціям оцінити поточну зрілість управління проектами та визначити галузі поліпшення;

- якщо організація приймає рішення розвивати практики управління проектами і переходити на нові більш високі рівні зрілості, то в справу вступає елемент «покращення» (improvement), який допомагає компаніям побудувати схему розвитку управління проектами таким чином, щоб забезпечити максимально ефективно досягнення своїх стратегічних цілей.

Основне призначення ОРМЗ – бути стандартом для корпоративного управління проектами та організаційної зрілості з управління проектами.

Основна відмінна риса ОРМЗ – це наявність унікальної бази даних, яка містить сотні кращих практик, опис тисяч ключових факторів успіху, результатів та іншої інформації, що характеризує розвиток зрілості управління проектами в організації.

ОРМЗ спроектований таким чином, щоб бути легким в розумінні і використанні, масштабується, гнучким і налаштованим на споживача. Грунтуючись на базі ОРМЗ як стандарту управління проектами, організація може успішно перейти до такого стану, коли проекти будуть досягати поставлених цілей в рамках бюджету, термінів і, що більш важливо, переслідуючи корпоративні стратегічні цілі [1].

Специфіка ІТ-проектів знаходить відображення також у специфічній методології управління проектами:

- CMMi;
- Microsoft Solution Framework;
- Rational Unified Process.

Контрольні питання?

- 1) Назвіть найважливіші положення при управлінні проектами?
- 2) Назвіть основні принципи нової концепції управління?
- 3) Які особливості при класифікації IT-проектів?
- 4) Назвіть особливості життєвого циклу проекту?
- 5) Основні вимоги до учасників проекту?
- 6) Як підібрати учасників в новий проект?
- 7) Які існують форми організаційної структури?
- 8) Назвіть успішні стандарти управління проектами?

2 Практичні аспекти управління ІТ-проектами

2.1 Процеси управління проектами

Розбиваючи життєвий цикл проекту на фази з проміжними результатами (рис. 2.1), ми, тим самим, робимо його більш керованим, знижуючи ступінь невизначеності від фази до фази. Більш того, деяка фаза може вилитися в окремий підпроект. Наприклад, на фазі визначення концепції проекту можуть знадобитися глибокі маркетингові дослідження ринку, що можна виділити в окремий проект маркетингового аналізу зі своїми власними фазами [1].

Для того, щоб провести проект по фазах до результату, необхідно виконати деякі серії дій. Причому в кожному проекті виконуються схожі процеси, які не залежать від предметної області. Такими загальними для всіх проектів процесами зі схожим змістом є ініціація, планування, виконання, управління і завершення проекту. Їх взаємозв'язок показана на рис. 5. Стрілками вказані напрямки потоків інформації.

Як видно, за процесом ініціації проекту слідує процес планування і виконання проекту. Процеси управління можуть повертатися до процесів планування, якщо не досягнуть кінцевого результату, що задовольняє цілі і обмеження проекту. Процеси завершення закривають проект.

Інтегративний характер управління проектом вимагає, щоб група процесів моніторингу та контролю взаємодіяла з іншими групами процесів, як показано на рис. 5. Процеси моніторингу і контролю здійснюються в той же самий час, що і процеси, що входять в інші групи процесів. Таким чином, на рис. 5 процес моніторингу та контролю зображений як «фонова» група процесів для інших чотирьох груп [1].

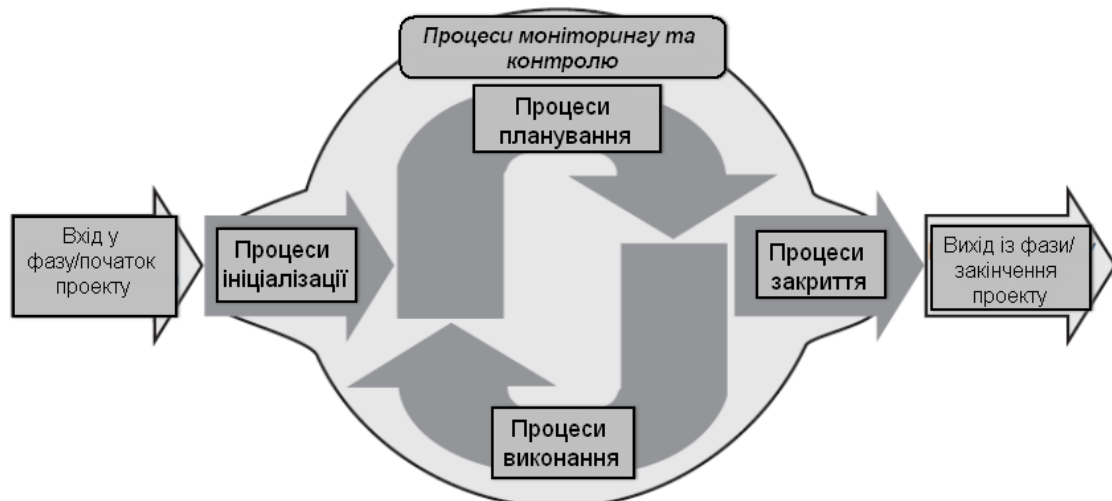


Рис. 2.1. Взаємозв'язок процесів управління

Ступінь взаємодії розглянутих груп процесів протягом життєвого циклу проекту різна. У реальності вони накладаються один на одного, як показано на рисунку 2.2.

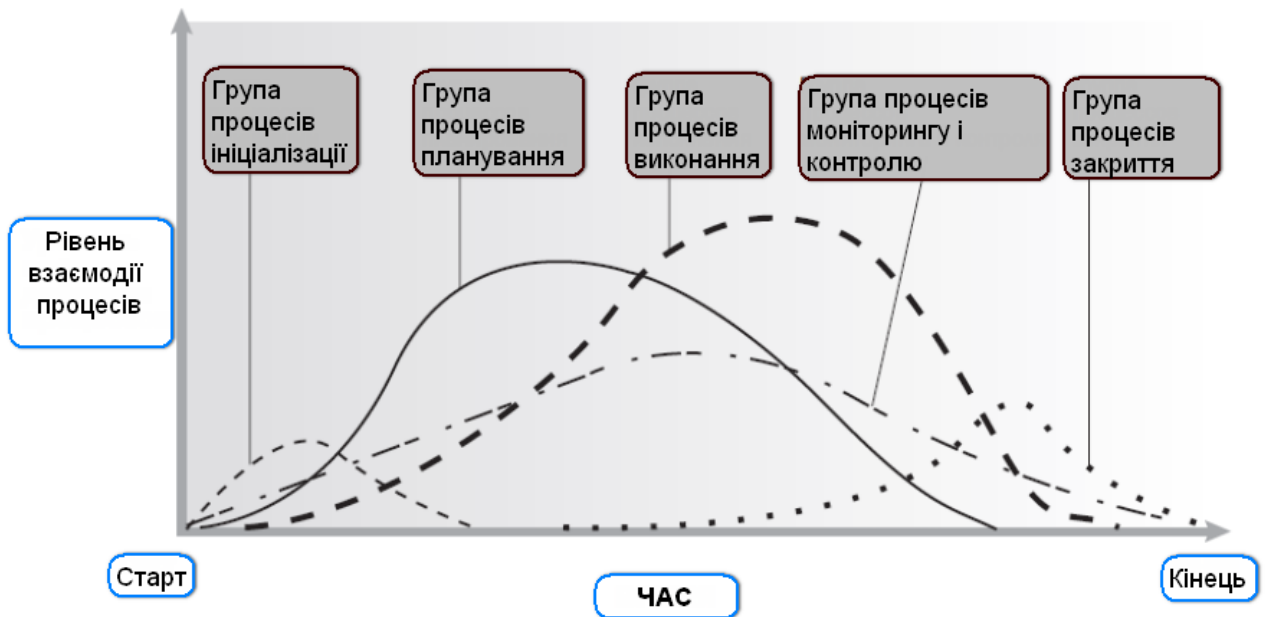


Рис. 2.2. Взаємодія груп процесів

Розкриємо зміст і взаємозв'язки кожної групи процесів. Групи процесів 2-5 покажемо схематично, як це прийнято в стандарті ANSI PMI PMBOK – рис. 2.3-2.6.

Охарактеризуємо зміст процесів управління проектами.

Ініціалізація (1) – визначення ділової потреби в проекті і його авторизація, а саме [1, 7]:

- вибір проекту і визначення ділових потреб;
- збір інформації;
- визначення цілей проекту;
- визначення обмежень і припущень;
- розробка опису продукту;
- визначення обов'язків менеджера проекту;
- визначення вимог до людських ресурсів (кадри, кваліфікація);
- оціночне визначення ресурсів;
- остаточне доопрацювання статуту проекту і призначення менеджера проекту.

Процеси планування (2) спрямовані на розробку планів по складовим проекту (розклад, вартість, бюджет, якість, персонал, ризики, взаємодія, контракти та ін.) І їх інтеграцію в цілісний, узгоджений документ – **План проекту**. Як показано на рис.5, планування – це процес, що постійно повторюється протягом усього життєвого циклу проекту [1].

У плануванні виділяють основні процеси, присутні завжди і виконуються в строго визначеному порядку, і допоміжні процеси, що залежать від характеру проекту. На рис. 2.3 показаний склад і зв'язки процесів планування. Коротко пояснимо кожен процес з рис. 2.3:

Планування змісту здійснюється на основі статуту проекту і інших вхідних документів, складається документ, що описує [1]:

- а) уточнений опис продукту і результатів поставки;
- б) класифікацію можливих змін і способів їх виявлення;
- в) порядок оцінки та включення змін в проект.

Визначення змісту – розбиття, декомпозиція цілей проекту на менші і більш керовані частини (підцілі). Глибина декомпозиції повинна забезпечувати можливість призначення закінчених груп робіт і виконавців на ці частини. Результатом визначення змісту є ІСР (Ієрархічна Структура Робіт), англ. WBS (Work Breakdown Structure) [1].

Визначення складу операцій – підготовка переліку всіх операцій, що виконуються за проектом, і уточнення ІСР. Операції, що не включені в уточнену ІСР, вважаються включеними в проект і не підлягають виконанню.

Планування ресурсів – визначення потреби (складу, кількості) в людських і матеріальних ресурсах, необхідних для виконання операцій проекту.

Визначення взаємозв'язків операцій – виявлення взаємозв'язків і взаємозалежностей операцій, побудова мережових діаграм робіт проекту. Частина операцій пов'язана між собою жорсткою логікою, інші операції можуть виконуватися в довільній послідовності, тобто пов'язані м'якою логікою [1].

Оцінка тривалості операцій – встановлення кількості одиниць часу на операції проекту, обчислення резервів часу і критичного шляху з мінімальною гнучкістю за часом.

Оцінка вартості – кількісна оцінка можливих витрат на ресурси, що залучаються, складання кошторису і плану управління вартістю.

Планування управління ризиками – встановлення підходу і заходів (коли, як і що робити) при загрозі або настання небажаних і незапланованих подій і відхилень, з метою їх запобігання або ефективного реагування.

Складання розкладу – аналіз даних про послідовності і тривалості операцій і необхідних ресурсах з метою створення розкладу виконання проекту.

Розробка бюджету – визначення кошторисної вартості по окремих пакетах робіт і проекту в цілому.

Розробка плану проекту - інтеграція даних попередніх процесів і складання узгодженого **Плану проекту** – в одному або кількох документах.

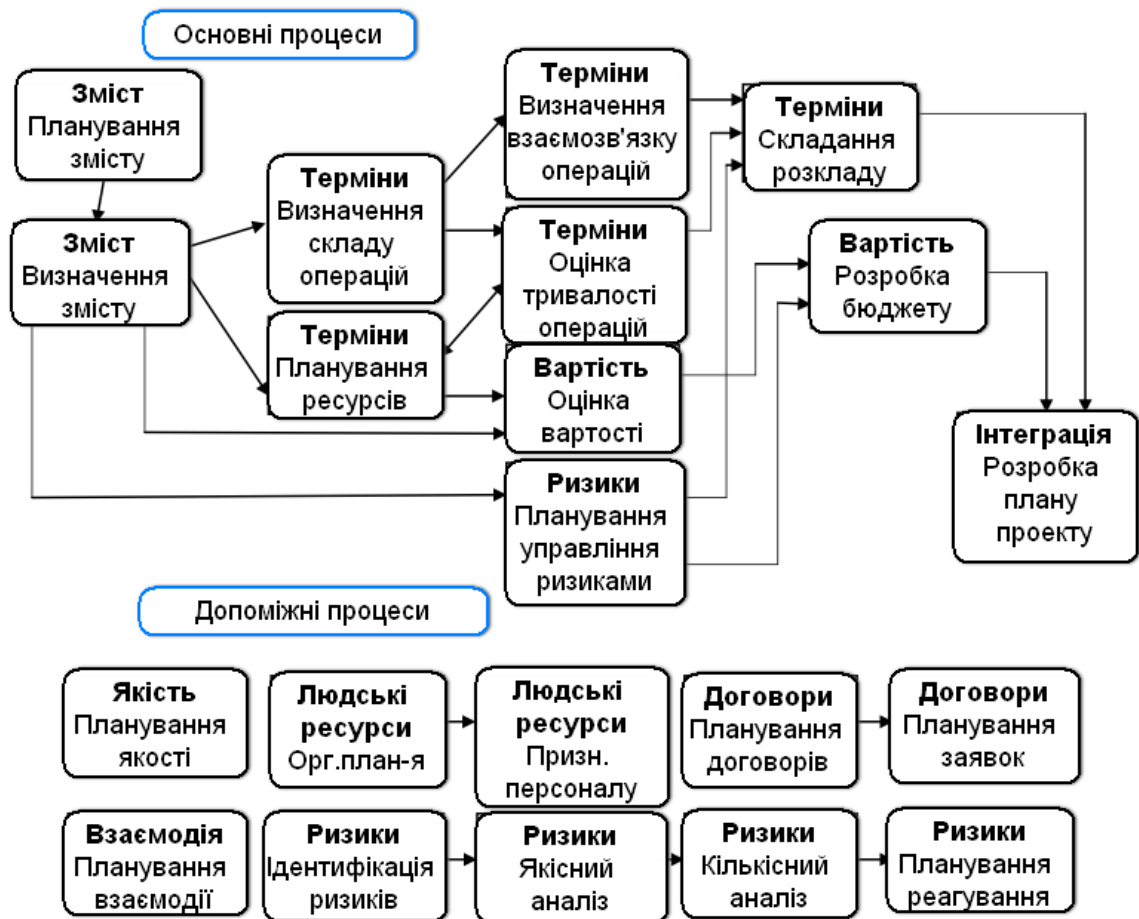


Рис. 2.3. Основні і допоміжні процеси

Допоміжні процеси планування встановлюють стандарти якості, розподіл ролей і відповідальності, інформаційні потреби учасників і способи взаємодії, виявляють ризики і наслідки їх впливу на цілі проекту і т.д.

Процеси виконання (3) показані на рис. 2.4. Виконання плану проекту – це безпосереднє виконання складових його операцій. Допоміжні процеси забезпечують гарантії якості, комплектацію/розподіл робіт та інформації, проведення нарад про хід робіт і ідентифікацію змін, розвиток навичок і знань команди, збір пропозицій постачальників, управління відносинами з постачальниками.

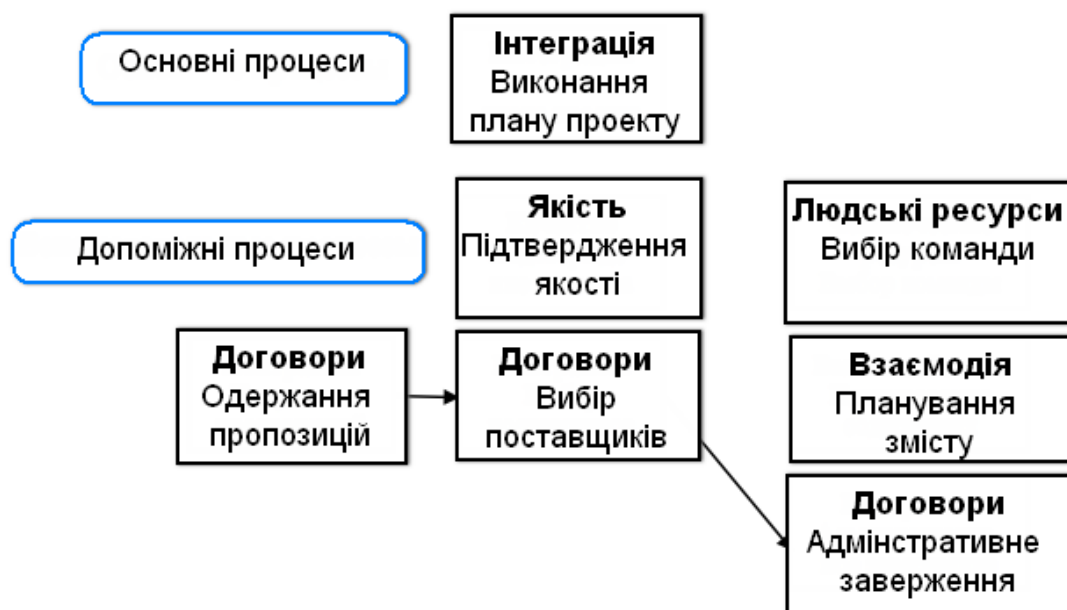


Рис. 2.4. Основні і допоміжні процеси 2

Процеси моніторингу, управління, аналізу (4) спрямовані на збір і розподіл звітності стану проекту, контроль відхилень, координацію змін розкладу та бюджету. Склад і зв'язки процесів виконання показані на рисунку 2.5.

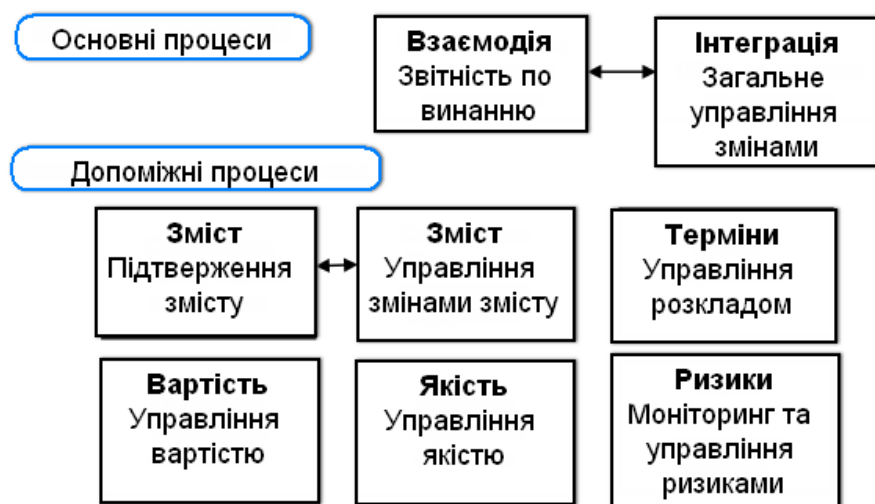


Рис. 2.5. Основні і допоміжні процеси 3

Звітність по виконанню – є збір і поширення інформації про хід проекту і прогнозах.

Загальне управління змінами – є координація змін по проекту в цілому.

Допоміжні процеси забезпечують посвідчення правильності виконання робіт, фіксацію і прийняття змін, контроль і зміну розкладу і бюджету, відповідність стандартам якості і усунення причин її зниження, відстеження та виявлення ризиків, оцінку заходів щодо зниження ризиків [1, 13].

Процеси завершення (5) впорядковуюють закриття проекту і складаються з процесів закриття контрактів (договорів) і адміністративного завершення (рис. 2.6), а саме [1]:

- перевірка і тестування кінцевого продукту;
- остаточні розрахунки з усіма учасниками проекту, фінансове закриття;
- остаточне оновлення документів проекту;
- завершення звіту про виконання проекту;
- збір, інтеграція накопичених знань і формування архіву проекту;
- офіційне приймання проекту замовником, передача і запуск в експлуатацію;
- звільнення задіяних ресурсів.

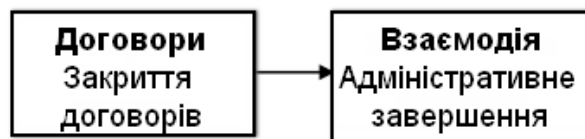


Рис. 2.6. Основні і допоміжні процеси 4

Грамотне та ефективне виконання перерахованих процесів вимагає від менеджера проекту знань в наступних областях [1]:

1. Управління інтеграцією проекту.
2. Управління змістом проекту.
3. Управління термінами проекту.
4. Управління вартістю проекту.
5. Управління якістю проекту.
6. Управління людськими ресурсами проекту.
7. Управління комунікаціями проекту.
8. Управління ризиками проекту.
9. Управління закупівлями проекту.
10. Управління зацікавленими особами.

2.2 Моделі управління проектами

Традиційна каскадна модель (англ. Waterfall model, іноді перекладають як модель «Водоспад») – модель процесу розробки програмного забезпечення, в якій процес розробки виглядає як потік, що послідовно проходить фази аналізу вимог, проектування, реалізації, тестування, інтеграції та підтримки. Як джерело назви часто вказують статтю, опубліковану У. У. Ройсом (W. W. Royce) в 1970 році; при тому, що сам Ройс використовував ітеративну модель розробки [1, 16].

У 1970 році в своїй статті Ройс описав у вигляді концепції те, що зараз прийнято називати «каскадна модель», і обговорював недоліки цієї моделі. Там же він показав як ця модель може бути доопрацьована до ітеративної моделі.

В оригінальній каскадній моделі Ройса, такі фази йшли в такому порядку [1]:

- визначення вимог;
- проектування;
- конструювання (також «реалізація» або «кодування»);
- втілення;
- тестування та налагодження (також «верифікація»);
- інсталяція;
- підтримка.

Перехід від однієї фази до іншої відбувається тільки після повного і успішного завершення попередньої

Дотримуючись каскадній моделі, розробник переходить від однієї стадії до іншої строго послідовно. Спочатку повністю завершується етап «визначення вимог», в результаті чого виходить список вимог до програмного забезпечення. Після того як вимоги повністю визначені, відбувається перехід до проектування, в ході якого створюються документи, що детально описують для програмістів спосіб і план реалізації зазначених вимог. Після того як проектування повністю виконане, програмістами виконується реалізація отриманого проекту. На наступній стадії процесу відбувається інтеграція окремих компонентів, що розробляються різними командами програмістів. Після того як реалізація і інтеграція завершені, проводиться тестування і налагодження продукту; на цій стадії усуваються всі недоліки, що з'явилися на попередніх стадіях розробки. Після цього програмний продукт впроваджується і забезпечується його підтримка – внесення нової функціональності і усунення помилок [1, 16].

Тим самим, каскадна модель (рис. 2.7) має на увазі, що перехід від однієї фази розробки до іншої відбувається тільки після повного і успішного завершення попередньої фази, і що переходів назад або вперед або перекриття фаз не відбувається.

Проте, існують модифіковані каскадні моделі (включаючи модель самого Ройса), що мають невеликі або навіть значні варіації описаного процес.

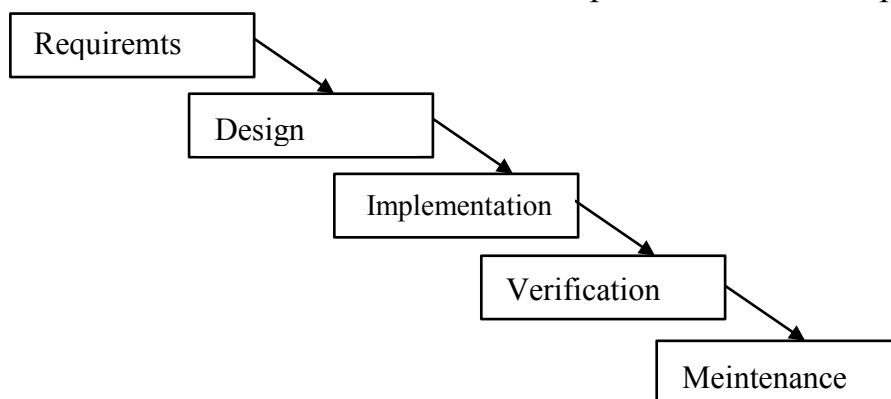


Рис. 2.7. Каскадна модель управління проектами

Методику «Каскадна модель» досить часто критикують за недостатню гнучкість і оголошення самоціллю формальне управління проектом на шкоду

термінів, вартості та якості. Проте, при управлінні великими проектами формалізація часто була дуже великою цінністю, так як могла кардинально знизити багато ризиків проекту і зробити його більш прозорим. Тому навіть в РМВОК 3-ої версії формально була закріплена лише методика «Каскадної моделі» і не були запропоновані альтернативні варіанти [1].

Починаючи з РМВОК 4-й версії вдалося досягти компромісу між методологіями, прихильними формальному і поступальному управлінню проектом, з методологіями, що роблять ставку на гнучкі ітеративні методи. Таким чином, починаючи з 2009 року, формально Інститутом управління проектами (PMI) пропонується як стандарт гібридний варіант методології управління проектами, що поєднує в собі як плюси від методики «Водограю», так і досягнення ітеративних методологів [16].

PRojects IN Controlled Environments 2 (PRINCE2) являє собою структурований метод управління проектами, схвалений урядом Великобританії в якості стандарту управління проектами в соціальній сфері. Методологія PRINCE2 включає в себе підходи до менеджменту, контролю та організації проектів.

Спочатку метод був розроблений в 1989 році Central Computer and Telecommunications Agency (CCTA) у Великобританії як стандарт для керівництва проектами в сфері інформаційних технологій. В даний час широко використовується і є «de facto» стандартом для керівництва проектами в Великобританії.

Наведемо переваги даного підходу. PRINCE2 є структурований підхід до управління проектами, тобто це метод для управління проектами в рамках чітко визначеної структури. PRINCE2 описує процедури для координації діяльності команди проекту при розробці і контролю над проектом, а також процедури, які використовуються при зміні проекту або якщо є істотні відхилення від початкового плану. У методі кожен процес визначається зі своїми основними входами і виходами, і з конкретними цілями і заходами, які будуть здійснюватися, що дає автоматичний контроль будь-яких відхилень від плану. За рахунок поділу процесів на керовані етапи, метод дає можливість ефективного управління ресурсами [1].

До недоліків можна віднести відсутність будь-якого регламентування з боку методології підходів до управління контрактами поставок, учасниками проекту та іншими процесами, які були винесені розробниками за рамки. Вважається, що кожен менеджер проекту вибирає власні методи і підходи до цієї роботи.

Діаграма (рис. 2.8) показує процеси методу PRINCE2. Стрілки показують напрямки інформаційних потоків.

Початок проекту. Як від початкової ідеї проекту (відображеної в мандаті на проект) перейти до безпосередньої реалізації цих ідей. Створюється організація – мінімум призначаються керівник проекту і Голова Комітету проекту. Формулюється короткий опис проекту (project brief) і підхід до його реалізації. Детально планується стадія запуску проекту.

Ініціація проекту. Проводиться планування проекту, включаючи план якості. Створюється економічне обґрунтування проекту (Business Case) і відкривається журнал ризиків, проводиться оцінка ризиків проекту. Плануються віхи, точки контролю проекту [1].

Управління проектом. Тут зосереджені ворота прийняття рішень Комітетом проекту (в тому числі щодо дострокового завершення проекту) та ситуаційне управління по значним проблемам і відхиленням.

Контроль стадій. Безпосередня робота керівника проекту по щоденному управлінню проектом – видача і приймання завдань, фіксація складнощів і ризиків, прийняття рішення про ескалацію, звітність перед Комітетом.

Управління виробництвом продукту. Заходи, які виконавці та робочі групи повинні зробити для визначення обсягів роботи, звіти про прогрес і передачі виконаної роботи [1].

Контроль кордонів стадій. Тут відбувається аналіз виконання плану стадії, проміжне планування наступної стадії, запасних планів, огляд ризиків та бізнес-плану. Служить для переходу між стадіями.

Завершення проекту. Як закрити проект, як управляти наступними діями, як розбирати огляди переваг проекту.

Планування. Як планувати, незалежно від того, коли здійснюється планування.

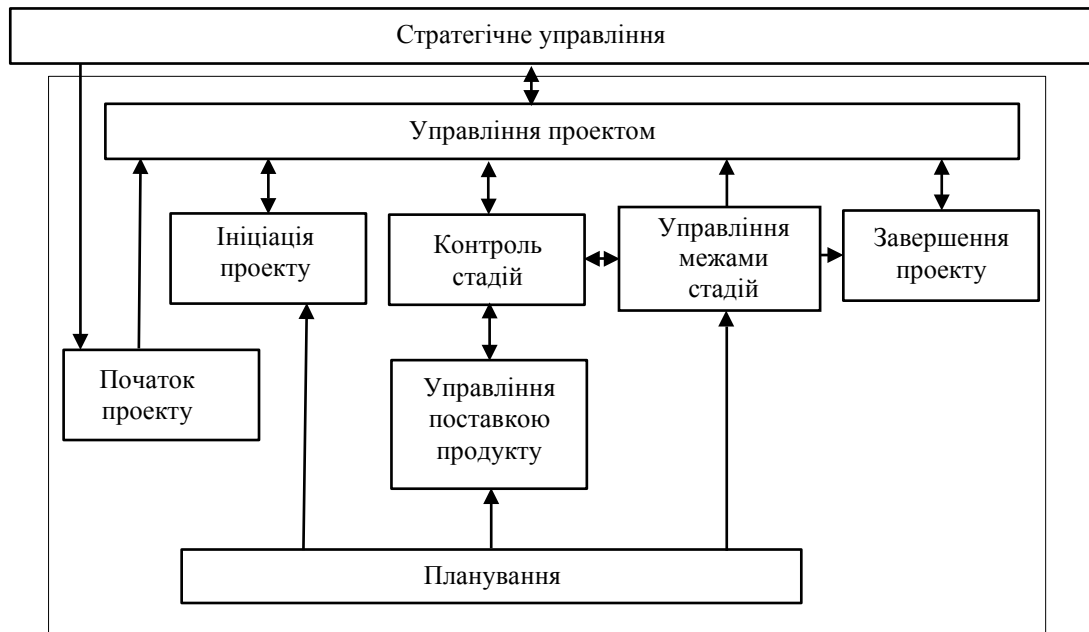


Рис. 2.8. Діаграма процесів методу PRINCE2

Організаційна структура:

- Менеджер проекту в традиційному розумінні;
- Комітет проекту (project board), перед яким регулярно звітує менеджер. Складається з 3х чоловік – замовника, головного користувача і головного спеціаліста. Рада проекту відповідальна за прийняття стратегічних рішень. Менеджер проекту зобов'язаний відстежувати можливі проблеми і пропонувати раді альтернативні рішення. Рада вирішує – який шлях краще [1].

– служба project assurance (аналог проектного офісу), мета якої надавати незалежну думку про проект з точки зору тих же трьох груп людей – замовників, користувачів і фахівців (в предметній області). Служба готує три звіти [1]:

- 1) business report (звіт про фінансовий стан проекту і вигідності проекту в цілому),
- 2) user report (наскільки добре виконуються вимоги користувачів),
- 3) technical report (наскільки хороший проект в технологічному плані - чи туди він рухається).

– Служба адміністративної підтримки (адміністратори проектів і т. п.), відповідальна за проведення зустрічей, доведення потрібної інформації до всіх її адресатів, збереження проектною інформації і т. п. У разі маленьких проектів це робить менеджер проекту.

Гнучка методологія розробки (англ. Agile software development, agile – методи) – серія підходів до розробки програмного забезпечення, орієнтованих на використання ітеративної розробки, динамічне формування вимог і забезпечення їх реалізації в результаті постійної взаємодії всередині самоорганізованих робочих груп, що складаються з фахівців різного профілю. Існує кілька методик, що відносяться до класу гнучких методологій розробки, зокрема екстремальне програмування, DSDM, Scrum, FDD [1, 14].

Застосовується як ефективна практика організації праці невеликих груп (які роблять однорідну творчу роботу) в поєднанні з керуванням ними комбінованим (ліберальним і демократичним) методом.

Більшість гнучких методологій націлені на мінімізацію ризиків шляхом зведення розробки до серії коротких циклів, званих ітераціями, які зазвичай тривають два-три тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі і включає всі завдання, необхідні для видачі міні-приросту по функціональності: планування, аналіз вимог, проектування, програмування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі, що гнучкий програмний проект готовий до випуску в кінці кожної ітерації. Після закінчення кожної ітерації команда виконує переоцінку пріоритетів розробки [1].

Agile-методи роблять акцент на безпосереднє спілкування віч-на-віч. Більшість agile-команд розташовані в одному офісі. Як мінімум, вона включає і «замовників» (англ. Product owner – замовник або його повноважний представник, який визначає вимоги до продукту; цю роль може виконувати менеджер проекту, бізнес-аналітик або клієнт). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних редакторів і менеджерів.

Основною метрикою agile-методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це призвело до критики цих методів як недисциплінованих.

У лютому 2001 в штаті Юта США був випущений «Маніфест гнучкої методології розробки програмного забезпечення». Він був альтернативою керованим документацією, «великоваговим» практикам розробки програмного забез-

печення, таким як «метод водоспаду», який був золотим стандартом розробки в той час. Даний маніфест був схвалений і підписаний представниками методологій: екстремального програмування, Crystal Clear, DSDM, Feature driven development, Scrum, Adaptive software development, Pragmatic Programming. Гнучка методологія розробки використовувалася багатьма компаніями і до прийняття маніфесту, однак входження Agile-розробки в маси відбулося саме після цієї події [1].

Agile – сімейство процесів розробки, а не єдиний підхід в розробці програмного забезпечення, і визначається Agile Manifesto. Agile не включає практик, а визначає цінності і принципи, якими керуються успішні команди.

Agile Manifesto розроблений і прийнятий 11-13 лютого 2001 року на лижному курорті The Lodge at Snowbird в горах Юти. Agile Manifesto містить 4 основні ідеї та 12 принципів. Потрібно відмітити, що Agile Manifesto не містить практичних порад.

Основні ідеї [1]:

- люди і взаємодія важливіше процесів та інструментів;
- працюючий продукт важливіше вичерпної документації;
- співпраця з замовником важливіше узгодження умов контракту;
- готовність до змін важливіше ніж дотримання початкового плану.

Принципи, які роз'яснює Agile Manifesto:

- задовольнити клієнта за рахунок ранньої і безперебійної поставки цінного програмного забезпечення;
- готовність до змін вимог навіть в кінці розробки (це може підвищити конкурентоспроможність отриманого продукту);
- часта поставка робочого програмного забезпечення (кожного місяця або тижня або ще частіше);
- тісне, щоденне спілкування замовника з розробниками протягом усього проекту;
- проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;
- рекомендований метод передачі інформації – особиста розмова (обличчям до обличчя);
- працююче програмне забезпечення – кращий вимірник прогресу;
- спонсори, розробники і користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійна увага поліпшенню технічної майстерності і зручного дизайну;
- простота – мистецтво не робити зайвої роботи;
- кращі технічні вимоги, дизайн та архітектура виходять у самоорганізованій команді;
- постійна адаптація до обставин, що змінюються.

Один із повторюваних пунктів критики: при agile-підході часто нехтують створенням плану («дорожньої карти») розвитку продукту, так само як і управ-

лінням вимогами, в процесі якого і формується така «карта». Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість замовника раптом і несподівано в кінці кожної ітерації виставляти нові вимоги, часто суперечать архітектурі вже створеного і поставленого товару. Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації [1].

Крім того, вважається, що робота в Agile мотивує розробників вирішувати всі задачі, що надійшли, найпростішим і найшвидшим із можливих способів, при цьому часто не звертаючи уваги на правильність коду з точки зору вимог платформи (підхід - «працює, і добре», при цьому не враховується, що може перестати працювати при найменшій зміні або ж дати важкі до відтворення дефекти після реального розгортання у клієнта). Це призводить до зниження якості продукту і накопичення дефектів.

RAD (від англ. Rapid application development – **швидка розробка додатків**) – концепція створення засобів розробки програмних продуктів, що приділяє особливу увагу швидкості і зручності програмування, створення технологічного процесу, що дозволяє програмісту максимально швидко створювати комп'ютерні програми. Практичне визначення: RAD –це життєвий цикл процесу проектування, створений для досягнення більш високої швидкості розробки і якості ПЗ, ніж це можливо при традиційному підході до проектування. З кінця ХХ століття RAD набула широкого поширення і схвалення. Концепцію RAD також часто пов'язують з концепцією візуального програмування [1, 17].

Концепція RAD стала відповіддю на недосконалі методи розробки програм 1970-х і початку 1980-х років, такі як «модель водоспаду» (англ. Waterfall model). Ці методи передбачали настільки повільний процес створення програми, що часто навіть вимоги до програми встигали змінитися до закінчення розробки. Засновником RAD вважається співробітник IBM Джеймс Мартін, який в 1980-х роках сформулював основні принципи RAD, ґрунтуючись на ідеях Баррі Бойм і Скотта Шульца. А в 1991 році Мартін опублікував відому книгу, в якій детально виклав концепцію RAD і можливості її застосування. В даний час RAD стає загальноприйнятною схемою для створення засобів розробки програмних продуктів [1].

RAD передбачає, що розробка ПЗ здійснюється невеликою командою розробників за термін близько трьох-чотирьох місяців шляхом використання інкрементного прототипування із застосуванням інструментальних засобів візуального моделювання та розробки. Технологія RAD передбачає активне залучення замовника вже на ранніх стадіях – обстеження організації, вироблення вимог до системи. Останнє із зазначених властивостей передбачає повне виконання вимог замовника як функціональних, так і не функціональних, з урахуванням їх можливих змін в період розробки системи, а також отримання якісної документації, що забезпечує зручність експлуатації і супроводу системи. Це означає, що додаткові витрати на супровід відразу після поставки будуть знач-

но менше. Таким чином, повний час від початку розробки до отримання прийнятного продукту при використанні цього методу значно скорочується [17].

Технологію RAD доцільно застосовувати, коли чітко визначені деякі пріоритетні напрямки розробки проекту [1].

1. Необхідно виконання проекту в стислі терміни. Швидке виконання проекту дозволяє створити систему, що відповідає вимогам сьогодення. Якщо система проектується довго, то дуже висока ймовірність, що за цей час суттєво зміняться фундаментальні положення, що регламентують діяльність організації, тобто, система морально застаріє ще до завершення її проектування.

2. Нечітко визначені вимоги до ПЗ. У більшості випадків замовник дуже приблизно уявляє собі роботу майбутнього програмного продукту і не може чітко сформулювати всі вимоги до ПЗ. Вимоги можуть бути взагалі не визначені до початку проекту або можуть змінюватися по ходу його виконання.

3. Проект виконується в умовах обмеженості бюджету. Розробка ведеться невеликими RAD-групами в короткі терміни, що забезпечує мінімум трудовитрат і дозволяє вписатися в бюджетні обмеження.

4. Інтерфейс користувача (GUI) є головний фактор. Немає сенсу змушувати користувача малювати картинки. RAD-технологія дає можливість продемонструвати інтерфейс в прототипі, причому досить скоро після початку проекту.

5. Можливо розбиття проекту на функціональні компоненти. Якщо передбачувана система велика, необхідно, щоб її можна було розбити на дрібні частини, кожна з яких має чітку функціональність. Вони можуть випускатися послідовно або паралельно (в останньому випадку залучається кілька RAD-груп).

6. Низька обчислювальна складність ПЗ.

RAD-технологія не є універсальною, тобто її застосування доцільно не завжди. Наприклад, в проектах, де вимоги до програмного продукту чітко визначені і не повинні змінюватися, залучення замовника в процес розробки не потрібно і більш ефективною може бути ієрархічна розробка (каскадний метод). Те ж стосується проектів, складність яких визначається необхідністю реалізації складних алгоритмів, а роль і обсяг призначеного для користувача інтерфейсу невеликий [1].

Принципи RAD технології спрямовані на забезпечення трьох основних її переваг – високій швидкості розробки, низької вартості і високої якості. Досягти високої якості програмного продукту досить непросто і одна з головних причин виникаючих труднощів полягає в тому, що розробник і замовник бачать предмет розробки (ПЗ) по-різному [17].

- Інструментарій повинен бути націлений на мінімізацію часу розробки;
- Створення прототипу для уточнення вимог замовника;
- Циклічність розробки: кожна нова версія продукту ґрунтується на оцінці результату роботи попередньої версії замовником;
- Мінімізація часу розробки версії, за рахунок перенесення вже готових модулів і додавання функціональності в нову версію;

- Команда розробників повинна тісно співпрацювати, кожен учасник повинен бути готовий виконувати кілька обов'язків;
- Управління проектом має мінімізувати тривалість циклу розробки.

Принципи RAD (рис. 2.9) застосовуються не тільки при реалізації, а й поширюються на всі етапи життєвого циклу, зокрема на етап обстеження організації, побудови вимог, аналіз і дизайн.

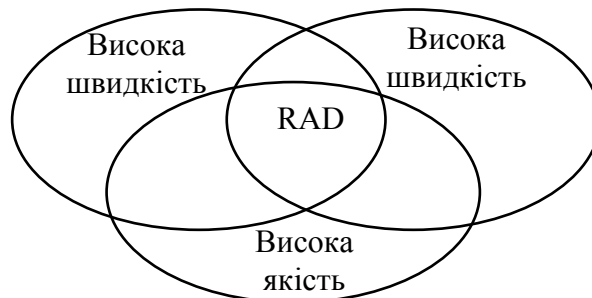


Рис. 2.9. Принципи RAD

Планування – сукупність вимог, отриманих при системному плануванні та аналізі процедури розробки життєвого циклу (SDLC). На цьому етапі користувачі, менеджери і IT-фахівці обговорюють завдання проекту, його обсяг, системні вимоги, а також складності, які можуть виникнути при розробці. Фаза завершується узгодженням ключових моментів з RAD-групою і отриманням від керівників проекту дозволу на продовження [1, 17].

Проектування користувачем – під час цього етапу користувачі, взаємодіючи з системними аналітиками, розробляють моделі і прототипи, які включають в себе всі необхідні системні функції. Для перекладу прототипів користувачів у робочі моделі RAD-група зазвичай використовує техніку об'єднаної розробки додатків (JAD) і CASE-інструменти. Проектування користувачем виявляється тривалим інтерактивним процесом, який дозволяє користувачам зрозуміти, змінити і в кінцевому рахунку вибрати робочу модель, що відповідає їх вимогам.

Конструювання – етап, в якому основне завдання полягає в розробці програм і додатків. Аналогічна стадії «реалізація» в SDLC. У RAD, однак, користувачі продовжують брати участь і, як і раніше, можуть пропонувати зміни або поліпшення у вигляді розроблених ними доповідей. В їх завдання входить програмування і розробка додатків, написання коду, інтеграція модулів і системне тестування.

Перемикання – включає в себе операції по конверсії даних, тестування, перехід на нову систему і тренування користувачів. За своїм завданням нагадує фінальну стадію SDLC. Порівнюючи з традиційними методами розробки ПЗ, весь процес виявляється стислим за часом. Як результат, нова система виявляється швидше побудованою, доставленою до замовника і встановленою на робочих місцях [1].



Рис. 2.10. Модель швидкої розробки RAD

Технологія швидкої розробки додатків (RAD) дозволяє забезпечити:

- швидкість просування програмного продукту на ринок;
- інтерфейс, що влаштовує користувача;
- легку адаптованість проекту до мінливих вимог;
- простоту розвитку функціональності системи.

2.3 Документація проекту

Якщо ви поговорите з досвідченим керівником проекту, то він напевно дасть вам чимало порад про те, що вам варто, а чого не варто робити для успішного управління будь-яким проектом. В основному всі вони будуть про те, як управляти людьми і їх роботою (а також і вашою роботою), як надавати результати вчасно і згідно з бюджетом, при цьому тримати рівень ризику на мінімумі [1, 18].

1. Статут проекту.

Наприклад, на початку вашого проекту вам знадобиться створити лише односторінковий документ – статут проекту. Даний документ повинен забезпечити те, що ви і ваш клієнт розуміють основні цілі проекту. Якщо ви не знаєте, куди йдете, як же ви зрозумієте, що ви біля цілі?

Для отримання даної інформації вам варто влаштувати зустріч з вашим клієнтом і задати наступні 3 важливих питання:

- Які цілі вашого проекту?
- Що ви хочете виробляти або надавати?
- Яке бізнес-обґрунтування для виконання даного проекту?

Після зустрічі запишіть всі відповіді на дані питання в вашому статуті проекту і перешліть його клієнту для отримання підтвердження.

Тепер ви виконали найбільш важливу частину вашого проекту, тобто усвідомили і погодили з вашим клієнтом призначення вашого проекту.

Якщо ви проаналізуєте час, витрачений на виконання даного кроку у вашому проекті, то у вас вийде приблизно дві зустрічі і 30 хвилин для запису всієї інформації, тобто приблизно дві години на маленький проект. Не так вже й багато [18].

2. План проекту

Тепер ви можете створити другий документ – план проекту. Він повинен включати в себе список всієї роботи, яка повинна бути виконана (іншими словами, масштаб проекту), хто її виконуватиме, витрати і час на виконання даної роботи і нарешті простий огляд того, що може піти не так, або ж дослідження ризиків.

3. Звіт про прогрес

Зазвичай раз в тиждень або на місяць вам необхідно надавати звіт про прогрес вашим клієнтам. Вони хочуть знати, що було виконано, як і скільки часу і коштів було витрачено. Вони також хочуть знати про те, чи необхідна вам їх допомога у вирішенні проблем. У ваші обов'язки входить збір цієї інформації, щоб ви могли створити даний документ.

Існує чотири основних типи документації на ПЗ [1]:

- архітектурна/проектна – огляд програмного забезпечення, що включає опис робочого середовища і принципів, які повинні бути використані при створенні ПЗ;
- технічна – документація на код, алгоритми, інтерфейси, API;
- для користувача – інструкції для кінцевих користувачів, адміністраторів системи та іншого персоналу;
- маркетингова.

Проектна документація зазвичай описує продукт в загальних рисах. Не описуючи того, як що-небудь буде використовуватися, вона відповідає на питання «Чому саме так». Наприклад, в проектному документі програміст може описати обґрунтування того, чому структури даних організовані саме таким чином. Описуються причини, чому який-небудь клас сконструйований певним чином, виділяються патерни, в деяких випадках навіть даються ідеї, як можна буде виконати поліпшення в подальшому. Нічого з цього не входить в технічну або документацію для користувача, але все це дійсно важливо для проекту [1].

При створенні програми, одного лише коду, як правило, недостатньо. Повинен бути наданий певний текст, що описує різні аспекти того, що саме робить код. Така **технічна документація** часто включається безпосередньо в вихідний код або надається разом з ним.

Подібна документація має сильно виражений технічний характер і в основному використовується для визначення і опису API, структур даних і алгоритмів.

Часто при створенні технічної документації використовуються автоматизовані засоби – генератори документації, такі як Doxygen, javadoc, NDoc і інші. Вони отримують інформацію зі спеціальним чином оформлених коментарів у вихідному коді, і створюють довідкові інструкції в будь-якому форматі, наприклад, у вигляді тексту або HTML.

Використання генераторів документації та документуючих коментарів багатьма програмістами визнається зручним засобом, з різних причин. Зокрема, при такому підході документація є частиною вихідного коду, і одні і ті ж інструменти можуть використовуватися для складання програми і одночасної збірки документації до неї. Це також спрощує підтримку документації в актуальному стані [1].

На відміну від технічної документації, сфокусованої на код і тому, як він працює, **призначена для користувача документація** описує лише те, як використовувати програму.

У разі якщо продуктом є програмна бібліотека, призначена для користувача документація та документація на код стають дуже близькими, майже еквівалентними поняттями. Але в загальному випадку, це не так.

Зазвичай, призначена для користувача документація являє собою інструкцію користувача, яка описує кожну функцію програми, а також кроки, які потрібно виконати для використання цієї функції. Хороша для користувача документація йде ще далі і надає інструкції про те, що робити в разі виникнення проблем. Дуже важливо, щоб документація не вводила в оману і була актуальною. Інструкція повинна мати чітку структуру; дуже корисно, якщо є наскрізний предметний покажчик. Логічний зв'язок і простота також мають велике значення [18].

Існує три підходи по організації документації для користувача. Довідник з основних функцій (англ. Tutorial), найбільш корисне для нових користувачів, послідовно проводить по ряду кроків, призначених для виконання певних типових задач. Тематичний підхід, при якому кожна глава інструкції присвячена якійсь окремій темі, більше підходить для користувачів, які вдосконалюються. В останньому, третьому підході, команди або завдання організовані у вигляді алфавітного довідника – часто це добре сприймається досвідченими користувачами, які добре знають, що вони шукають. Скарги користувачів зазвичай стосуються того, що документація охоплює тільки один з цих підходів, і тому добре підходить лише для одного класу користувачів [1].

У багатьох випадках розробники програмного продукту обмежують набір документації для користувача лише вбудованою системою допомоги (англ. Online help), що містить довідникову інформацію про команди або пункти меню. Робота по навчанню нових користувачів і підтримці вже існуючих перекладається на приватні видавництва, які часто надають значну допомогу розробникам.

Маркетингова документація – для багатьох додатків необхідно розташовувати поруч з ними рекламні матеріали, з тим, щоб зацікавити людей, звернувши їхню увагу на продукт. Така форма документації має на меті [1]:

- підігріти інтерес до продукту у потенційних користувачів;
- інформувати їх про те, що саме робить продукт, з тим, щоб їх очікування збігалися з тим, що вони отримують;
- пояснити положення продукту в порівнянні з конкуруючими рішеннями.

Одна з хороших маркетингових практик – надання слогана – простої фрази, що легко запам'ятовується та ілюструє те, що необхідно донести до користувача, а також характеризує відчуття, яке створює продукт.

Часто буває так, що коробка продукту та інші маркетингові матеріали дають більш ясну картину про можливості та способи використання програми, ніж все інше.

2.4 Управління ризиками проекту

Ризик – це невизначена подія або умова, настання якої негативно або позитивно позначається на цілі проекту, такі як зміст, розклад, вартість і якість. Причому вплив ризику на проект може бути, як негативним, так і позитивним. На практиці ризики розглядаються як загроза проектам.

Ризик може бути викликаний однією або декількома причинами і в разі виникнення може вплинути на один або кілька аспектів. Причиною може бути існуюча або потенційна вимога, припущення, обмеження або умова, яка створює ймовірність негативних або позитивних наслідків. Наприклад, причиною ризику може бути необхідність отримання дозвільної документації в галузі охорони навколишнього середовища або недолік персоналу, залученого для розробки проекту. Ризиком в першому випадку буде затримка з видачею дозволу контролюючим органом, а в другому, в разі сприятливої можливості, додатковий персонал, який може бути залучений до розробки проекту, може стати доступним для призначення на проект. Виникнення будь-якого з цих точно не відомих заздалегідь подій може вплинути на проект, його зміст, вартість, розклад, якість або виконання. До умов виникнення ризиків можуть також відноситись аспекти середовища організації або проекту, що сприяють збільшенню ризику (наприклад, незрілі практики управління проектами, відсутність загальних систем управління, одночасне виконання кількох проектів або залежність від зовнішніх учасників проекту, яких неможливо контролювати безпосередньо [1]).

Причини ризиків проекту знаходяться в невизначеності, яка присутня у всіх проектах. Відомі ризики – це ті ризики, які були ідентифіковані і проаналізовані, що дозволяє планувати реагування на них. Для тих відомих ризиків, якими неможливо управляти проактивно, слід виділити резерв на можливі втрати. Невідомими ризиками неможливо управляти проактивно, і, отже, для них можна виділити управлінський резерв. Існуючий негативний ризик проекту розглядається як проблема [1].

Окремі ризики проекту відрізняються від загального ризику проекту. Загальний ризик проекту відображає ефект невизначеності по відношенню до всього проекту. Це більше ніж сума окремих ризиків в проекті, так як сюди входять всі джерела невизначеності проекту. Він відображає схильність зацікавлених сторін впливу (як позитивного, так і негативного) від варіацій в кінцевому результаті проекту [1, 10].

Базові плани по утриманню, по термінам і по вартості створюються з урахуванням нейтралізації частини ризиків, передбачуваних з імовірністю 100%. Тому базові плани включають роботи і витрати, які належить виконати з імовірністю 100% [12].

Ризики, що не очікувані з імовірністю 100% і не припускають нейтралізації, не відображаються в базових планах. Наступ таких ризиків, призводить до наслідків, які, як правило, виливаються в додаткові роботи і витрати (грошей і часу), ймовірність яких до цього вважалася менше 100%. Такі ризики покриваються за рахунок резервів [1].

Класифікації ризиків. Ризики поділяють на відомі і невідомі. Відомі ризики – це ризики, які виявлені, ідентифіковані. Їх можна аналізувати і планувати. Невідомі – неідентифіковані ризики не піддаються управлінню, але можуть бути враховані при формуванні резервів.

З точки зору керованості, ризики поділяють на внутрішні і зовнішні. Внутрішні ризики – це події, умови та процеси, які команда проекту може контролювати. Зовнішні ризики – це події, умови та процеси, які виходять за межі впливу команди проекту. Наприклад, зміни законодавства країни, зміни вимог і пріоритетів спонсорів, зміни у виконуючій організації або у замовника, ринкові зміни, цивільні і природні катаклізми та інші форс-мажорні обставини [10].

Залежно від джерел виникнення ризики можна розділити на наступні категорії [1]:

- технічні ризики – пов'язані з помилками проектування, використання неперевіраних технологій, порушенням промислових стандартів та ін.;

- управлінські ризики – пов'язані з недоліками в плануванні та управлінні проектом на рівні менеджера проекту. Наприклад, невдало складений розклад, погано описані ролі і відповідальності, підбір недостатньо кваліфікованого персоналу, часті перестановки в команді, помилкові оцінки та розрахунки виконання проекту і т.д.;

- організаційні ризики – виникають через недоробки на рівні топ-менеджера проекту і пов'язані з неузгодженістю між проектами, низькою проектною дисципліною і конфліктами через ресурси, несумісністю цілей проекту, сильним впливом зовнішніх факторів, недостатнім або нестабільним фінансуванням і т.д.;

- ділові ризики – пов'язані зі змінами бізнес-середовища і бізнес-умов, в яких ініціювався проект. Наприклад, помилки в ринкових прогнозах, низька відповідальність провідних стейкхолдерів, зміна пріоритетів та вимог спонсора або замовника, інші ризики замовника та ін.;

- ризики навколишнього середовища – природно-кліматичні, екологічні та інші ризики;

- соціальні і політичні – страйки, державні перевороти і ін.;

- ризики зловмисних дій і т.д.

Процес виявлення та ідентифікації ризиків повинен бути безперервним і постійним протягом всіх фаз життєвого циклу проекту. Адже ризики можуть змінюватися, зникати, можуть бути виявлені нові, раніше невідомі ризики. У

міру просування проекту до завершення і зниження загальної невизначеності, велика частина ризиків повинна бути ідентифікована, оцінена, "переможена".

Вирішуючи завдання з управління ризиками, менеджер проекту і команда управління ризиками проекту, періодично проходять і повертаються до наступних дій [1]:

- ідентифікація та оцінка ризиків (2 основних параметри оцінки ризику – ймовірність виникнення і наслідки);
- мінімізація ймовірності настання ризиків, запобігання або підготовка до наступу ризиків;
- реагування на ризики, мінімізація негативних наслідків ризиків, а якщо ризик все ж настав, то знаходження способів подолання та забезпечення виконання проекту за планом;
- фіксація відхилень від базових планів щодо термінів, за вартістю і за змістом; якщо наслідки серйозні і відхилення від планів неминучі, зміна і узгодження планів з урахуванням наслідків, виконання коригувальних дій.

Таким чином, управління ризиками можна визначити, як систематичний процес ідентифікації, аналізу і реагування на проектні ризики.

Управління ризиками вимагає додаткових витрат. Ці витрати повинні порівнюватися з бюджетом всього проекту, бути необхідними і достатніми, щоб забезпечити досягнення цілей і результатів проекту. Суми витрат на управління ризиками в проектах можуть коливатися в діапазоні 1%-15% всього бюджету проекту [1,10].

Дії з управління ризиками робляться на етапах планування та управління. Ці дії залежать від характеру проекту і є допоміжними процесами. У стандарті ANSI PMBOK виділяють 6 складових процесів управління ризиками, причому перші п'ять з них спрямовані на попередню роботу над ризиками, на підготовку до виникнення ризиків [1]:

1. Планування управління ризиками;
2. Ідентифікація ризиків;
3. Якісний аналіз ризиків;
4. Кількісний аналіз ризиків;
5. Планування реагування на ризики;
6. Контроль ризиків.

Планування управління ризиками – процес, який визначає, яким чином здійснювати управління ризиками проекту. Ключова вигода даного процесу полягає в забезпеченні того, щоб ступінь, тип і наочність управління ризиками були відповідні ризикам і важливості проекту для організації [12].

Процес планування управління ризиками повинен визначити правила і підходи до управління ризиками проекту, а саме:

- способи виявлення та джерела ризиків;
- ролі і відповідальності, як на рівні керівництва, так і на рівні членів команди – може бути створена спеціальна команда управління ризиками;
- бюджет управління ризиками;
- періодичність ідентифікації, аналізу та оцінок ризиків;

- критерії порогових величин, після яких потрібне втручання;
- форми звітності і документування.

Розкриття перерахованих пунктів призводить до створення на виході процесу плану управління ризиками. На вході процесу необхідно мати такі джерела як [1]:

- план управління проектом;
- статут проекту;
- реєстр зацікавлених осіб;
- фактори середовища підприємства;
- активи процесів організації;
- інша необхідна та доступна інформація з плану проекту.

Корисним додатком до плану управління ризиками може бути побудова ієрархічної структури ризиків (ІС Ризиків). Джерела (причини) ризиків деталізуються до таких ризиків, за які можна призначити одну відповідальну особу (рис. 2.11).

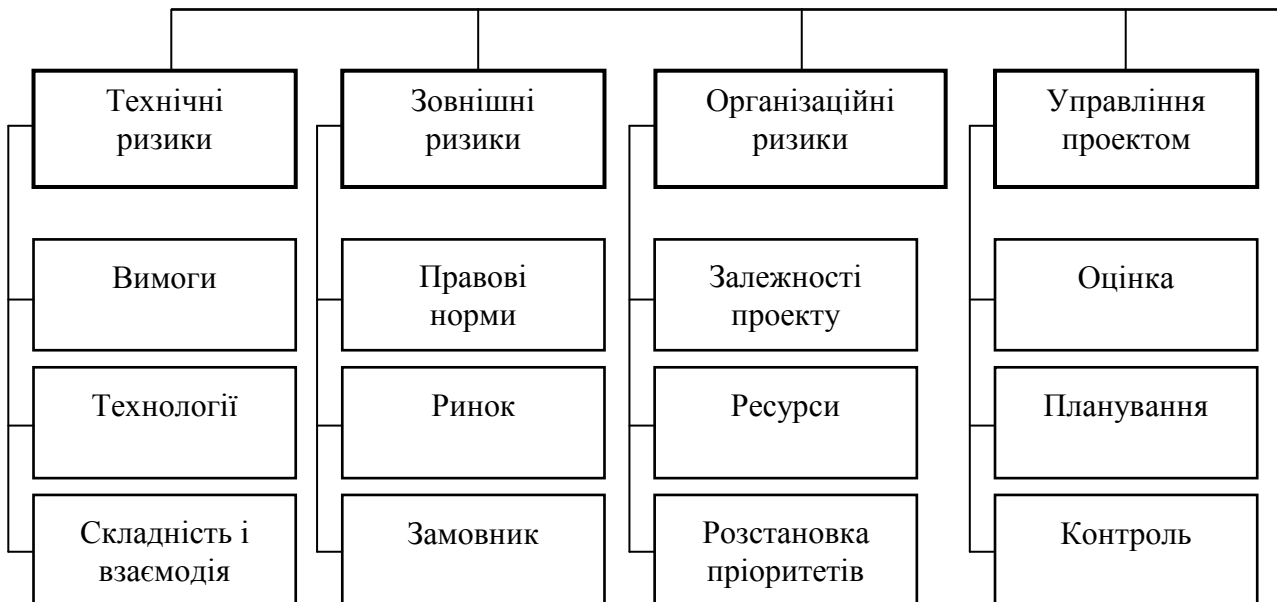


Рис. 2.11. Приклад ІС Ризиків

Ідентифікація ризиків – процес визначення переліку ризиків, які можуть впливати на проект, і документування їх характеристик. Ключова вигода даного процесу полягає в документуванні існуючих ризиків, а також в знаннях і можливостях, які це надає команді проекту для того, щоб передбачити можливі події [1].

На вході процесу ідентифікації ризиків передбачається мати максимальне число джерел даних [1]:

- план управління ризиками;
- план управління вартістю;
- план управління розкладом;
- план управління якістю;

- план управління людськими ресурсами;
- базовий план за змістом;
- оцінки вартості операцій;
- оцінки тривалості операцій;
- реєстр зацікавлених осіб;
- документи проекту;
- закупівельна документація;
- фактори середовища підприємства;
- активи процесів організації.

На виході необхідно отримати реєстр ризиків (пронумерований список ризиків з описом).

На цьому етапі ризики повинні бути чітко розмежовані і про ризики повинні бути точно записані такі дані:

- найменування і дата ідентифікації ризику;
- опис ризику.

По ходу виконання проекту інформація про ризики повинна оновлюватися і доповнюватися такими даними:

- особа, відповідальна за управління ризиком;
- посилання на ІСР, де можуть виникнути додаткові роботи;
- ймовірність виникнення ризику;
- наслідки ризику;
- стратегія реагування на ризик і т.д

Ідентифікація ризиків - це процес, що вимагає інтелектуальних зусиль, абстрагування, досвіду і знань.

Можливі інструменти і методи [1]:

- огляд документації;
- методи збору інформації:
 - 1) мозковий штурм;
 - 2) метод Дельфі;
 - 3) проведення інтерв'ю;
 - 4) аналіз першопричини;
- аналіз за допомогою контрольного списку;
- аналіз припущень;
- методи діаграм:
 - 1) діаграми причинно-наслідкових зв'язків;
 - 2) блок-схеми процесу і системи;
 - 3) діаграми впливу;
- аналіз SWOT;
- експертна оцінка.

Можна здійснювати структурований аналіз документації по проекту, включаючи плани, допущення, архіви попередніх проектів, угоди та іншу інформацію. Якість планів, а також узгодженість планів з вимогами і припущеннями проекту можуть служити показниками ризиків в проекті.

Метою **мозкового штурму** є створення всеосяжного списку ризиків проекту. Як правило, мозковий штурм проводить команда проекту, часто за участю ряду експертів з різних областей, які не є членами команди. Генерація ідей, що відносяться до ризиків проекту, відбувається під керівництвом модератора або в традиційній вільній формі мозкового штурму, або за допомогою структурованих методів проведення масових інтерв'ю. За основу може прийматися система категорій ризиків, наприклад ієрархічна структура ризиків. Далі ризики підлягають ідентифікації та категоризації за типами, а їх визначення – уточненню [1].

Метод Дельфі – це спосіб досягнення консенсусу між експертами. Даний метод передбачає, що експерти за ризиками проекту беруть в ньому участь анонімно. За допомогою опитувального листа модератор збирає ідеї про важливі ризики проекту. Відповіді резюмуються і потім повертаються експертам для подальших коментарів. Консенсусу можна досягти за кілька циклів даного процесу. Метод Дельфі допомагає знизити необ'єктивність в оцінці даних і усуває надмірний вплив окремих осіб на кінцевий результат [1].

Проведення інтерв'ю серед досвідчених учасників проекту, зацікавлених сторін або експертів по предметній області сприяє ідентифікації ризиків.

Аналіз першопричини являє собою особливий метод визначення проблеми, виявлення основних причин, що призвели до неї, і розробки запобіжних дій.

Контрольні списки ідентифікації ризиків розробляються на основі історичної інформації і знань, отриманих в ході виконання попередніх аналогічних проектів або з інших джерел інформації. В якості контрольного списку ризиків можна також використовувати найнижчий рівень RBS. Хоча контрольний список може бути коротким і простим, неможливо створити вичерпний список, і тому необхідно упевнитися, що контрольний список не використовується з метою уникнути зусиль по належній ідентифікації ризиків. Команда повинна також приділяти увагу питанням, які не знайшли свого відображення в контрольному списку. Крім того, контрольний список необхідно час від часу скорочувати для видалення або архівування пов'язаних пунктів. При завершенні проекту контрольний список слід переглядати, щоб врахувати в ньому уроки і поліпшити його для використання в майбутніх проектах [1].

Аналіз припущень досліджує обґрунтованість припущень стосовно проекту. Даний аналіз дозволяє ідентифікувати ризики проекту, що виникають внаслідок неточності, нестабільності, суперечливості чи неповноти припущень.

Діаграми причинно-наслідкових зв'язків, також відомі як діаграми Ісікави або діаграми «риб'ячий скелет», використовуються для визначення причин виникнення ризиків.

Блок-схеми процесу або системи – вид графічного відображення, демонструє порядок взаємодії різних елементів системи між собою і їх причинно-наслідкові зв'язки.

Діаграми впливу – графічне представлення ситуацій, що відображає причинно-наслідкові зв'язки, послідовності подій у часі та інші відношення між змінними і результатами.

Аналіз SWOT дозволяє провести аналіз проекту з точки зору кожного з аспектів: сильних і слабких сторін, сприятливих можливостей і загроз (strengths, weaknesses, opportunities, and threats, SWOT), що робить ідентифікацію ризиків більш повною, враховуючи ризики всередині проекту. При використанні даного методу починають з визначення сильних і слабких сторін організації, приділяючи особливу увагу або проекту, або організації, або області бізнесу в цілому. Потім в процесі аналізу SWOT ідентифікують будь-які сприятливі можливості проекту, обумовлені сильними сторонами організації, а також будь-які загрози, що з'являються внаслідок її слабких сторін. За допомогою даного аналізу також досліджують, наскільки сильні сторони організації компенсують загрози, і ідентифікують сприятливі можливості, які можна використовувати для подолання слабких сторін [1].

Ризики можуть бути ідентифіковані безпосередньо експертами, які мають відповідний досвід роботи в подібних проектах або областях бізнесу. Таких експертів повинен визначати керівник проекту і запрошувати для розгляду всіх аспектів проекту та ідентифікації можливих ризиків на основі свого попереднього досвіду і областей компетенції. Під час даного процесу необхідно враховувати необ'єктивність експертів.

Після того, як всі ризики ідентифіковані, проводять якісний аналіз ризиків з метою впорядкувати ризики за рівнями їх значимості.

Якісний аналіз ризиків – процес розстановки пріоритетів щодо ризиків для їх подальшого аналізу або дій, що виконується шляхом оцінки і зіставлення їх впливу та ймовірності виникнення. Ключова вигода даного процесу полягає в тому, що він дозволяє керівникам проектів зменшувати рівень невизначеності і фокусуватися на високопріоритетних ризиках. Значимість ризику визначається співвідношенням двох факторів – ймовірністю ризику і наслідками ризику для цілей проекту. При якісному аналізі ризиків ці два фактори описуються «оціночно», наприклад, ймовірність низька, середня, висока, наслідки незначні, помірні, значні і т.д. Тому таке ранжування ризиків за рівнями важливості не вимагає великих тимчасових і грошових витрат, великого обсягу докладної інформації [1].

На вході процесу якісного аналізу ризиків маємо:

- план управління ризиками;
- базовий план за змістом;
- реєстр ризиків;
- фактори середовища підприємства;
- активи процесів організації.

На виході маємо оновлення документів проекту: ранжування ризиків за рівнями важливості, перелік ризиків, які потребують додатково аналізу, тренд результатів при повторенні якісного аналізу, тобто тенденції зміни ризиків [7].

Основним методом якісного аналізу ризиків є використання матриці ймовірності і впливу. Вона містить дві шкали:

- шкалу ймовірності, яка зазвичай має лінійний діапазон значень [0,1], [1,10] або [1,100];

– шкалу наслідків, яка може бути, як лінійною, так і нелінійною, і відображає значимість наслідків.

Кожен ризик орієнтовно оцінюється за ймовірністю і наслідками і, згідно матриці, отримує певний ранг (рейтинг) важливості. Залежно від рангу, тобто клітини матриці, куди він потрапляє, ризики поділяють на низькі, середні і високі. Пороги для такого поділу ризиків в кожній організації встановлюються самостійно в залежності від толерантності до ризиків [1].

Крім того, в деяких організаціях можуть складатися окремі матриці ймовірності і наслідків для окремих цілей проекту (вартості, строків, змісту, якості) з різною шкалою і порогоми [1].

Оскільки точна оцінка ймовірності і наслідків на цьому етапі може бути ускладнена, то можна запропонувати варіант матриці ймовірності і наслідків зі спрощеної шкалою.

Кількісний аналіз ризиків зазвичай проводиться після якісного аналізу ризиків. Кількісний аналіз ризиків – процес чисельного аналізу впливу ідентифікованих ризиків на цілі проекту в цілому. Ключова вигода даного процесу полягає в тому, що він надає кількісну інформацію про ризики в підтримку процесу прийняття рішень з метою зменшення невизначеності проекту.

Цей процес полягає в наступному [1]:

- кількісній оцінці ймовірностей і наслідків кожного ризику, сортуванні ризиків за пріоритетами;
- визначенні ризиків, які потребують реагування та зосередження зусиль;
- кількісному визначенні величини резервів по вартості і термінам;
- визначенні найбільш реальних сценаріїв досягнення цілей за вартістю, терміном та змістом для проекту в цілому.

На вході процесу кількісного аналізу ризиків маємо як мінімум наступну інформацію:

- план управління ризиками;
- план управління вартістю;
- план управління розкладом;
- реєстр ризиків;
- фактори середовища підприємства;
- активи процесів організації.

На виході:

- поновлення документів проекту:
 - 1) перелік ризиків за пріоритетами, оцінка наслідків ризиків;
 - 2) імовірнісний аналіз проекту;
 - 3) тренди результатів (при повторному аналізі).

Кількісний аналіз ризиків проводиться щодо тих ризиків, які в результаті процесу якісного аналізу ризиків були класифіковані як потенційно і істотним чином впливаючі на конкуруючі вимоги проекту. У процесі кількісного аналізу ризиків оцінюється вплив цих ризиків на цілі проекту. Він використовується, в основному, для оцінки спільного впливу всіх ризиків на проект. Коли ризики

потрапляють в кількісний аналіз, даний процес може використовуватися для присвоєння числового рейтингу пріоритетності цих ризиків окремо [1].

У деяких випадках виконання процесу кількісного аналізу ризиків неможливо в зв'язку з відсутністю необхідних даних для розробки відповідних моделей. Керівник проекту повинен користуватися експертною оцінкою для визначення необхідності та доцільності кількісного аналізу ризиків. Вибір методу (методів) аналізу в кожному конкретному проекті визначається наявністю часу і бюджетом, а також потребою в якісному і кількісному описі ризиків і їх впливів. Щоб визначити, чи був ризик проекту успішно знижений, кількісний аналіз ризиків слід, при необхідності, повторно провести в рамках процесу контролю ризиків. Аналіз тенденцій може вказати на необхідність приділити більше або менше уваги відповідним діям з управління ризиками [1].

2.5 Програмні засоби для управління проектами

Впровадження управлінських інформаційних систем в організації сьогодні перестало бути лише засобом підвищення ефективності існуючої системи управління. Постійне вдосконалення методів управління організацією, що підкріплюється використанням сучасного програмного забезпечення є умовою успішного функціонування компанії на ринку. Розвиток інформаційних технологій постійно нагадує нам про закон переходу кількості в якість: бажане стає можливим, недоступне – доступним і економічно ефективним. Одним із завдань керівника стало крокувати в ногу з прогресом в інформаційних технологіях, щоб не відстати від конкурентів [1].

Microsoft Project є на сьогоднішній день найпоширенішою в світі системою УП (управління проектами). У багатьох західних компаніях MS Project став звичною добавкою до Microsoft Office навіть для рядових співробітників, які використовують його для планування графіків нескладних комплексів робіт.

Відмінною особливістю пакету є його простота. Розробники MS Project не прагнуть вкласти в пакет більш складні алгоритми календарного або ресурсного планування. У той же час значна увага приділяється використанню сучасних стандартів, що дозволяють ефективно інтегрувати пакет з іншими додатками.

Підтримка Microsoft Mail і Microsoft Exchange дозволяє полегшити і систематизувати групову роботу з проектами. Налаштування повідомлень для команди проекту включає можливість визначення складу проектних даних, які пересилаються учасникам проекту по електронній пошті і встановлення обмежень на корекцію інформації, яка пересилається одержувачами. Зберігання проектів в папках Exchange забезпечує додаткові засоби розмежування доступу до файлів проектів [1].

Для швидкого включення в роботу користувача-початківця MS Project надає, крім звичайних засобів допомоги, також можливість покрокової розробки проекту (Create Your First Project і Cue Cards) і інтелектуальної підказки (Answer Wizard). На жаль, Project поки не українізували, так що для ефективно-

го використання цих засобів необхідно знання англійської мови, включаючи специфічну термінологію управління проектами.

Серед переваг пакету також слід відзначити досить зручні і гнучкі засоби створення звітів. Основні типи звітів можуть бути обрані з заготовок (Report Gallery). Можливість одночасно мати до шести планів для кожного проекту дозволяє підвищити ефективність аналізу. У той же час MS Project надає мінімальний набір засобів для планування і управління ресурсами. Додаткові можливості Project також включають імпорт/експорт даних в форматах ASCII, CSV, Excel, Lotus 1-2-3, dBASE і FoxPro, засоби запису макрокоманд, Visual Basic [1].

MS Project може бути рекомендований для планування нескладних проектів користувачами непрофесіоналами і новачками.

Центральний програмний продукт сімейства Primavera, Primavera Project Planner, добре відомий в середовищі професійних менеджерів проектів в усьому світі. Сьогодні Project Planner застосовується для управління середніми і великими проектами в самих різних областях, хоча найбільше поширення даний продукт одержав у сфері управління будівельними та інженерними проектами.

Primavera Project Planner надає досить стандартний для всіх подібних систем графічний інтерфейс, але у нього є кілька додаткових можливостей. По-перше, це можливість групування і впорядкування робіт за різними ознаками на різних рівнях деталізації проекту, що дозволяє представити інформацію в більш зручному вигляді для конкретної управлінської ситуації. Наприклад, використовуючи дані кошти, всю інформацію по проекту можна згрупувати за фазою проекту на першому рівні ієрархії, по відповідному ресурсу на другому і відсортувати за датою початку робіт на третьому. Для кожної групи можуть бути задані власні шрифт і колір (тексту і фону), посторінкове розбиття.

Інша корисна особливість – це можливість розбиття екрану по горизонталі на дві частини, кожна з яких може бути переглянута незалежно. Це дає можливість одночасно переглядати різні частини проекту.

Крім того, Project Planner має певні відмінності від інших пакетів в засобах ресурсного планування.

При описі ресурсу можуть бути вказані нормальна і максимальна кількість наявності даного ресурсу, а також його ціна по шести тимчасових інтервалах. Ресурс може бути позначений як керуючий (обсяг призначення керуючого ресурсу на завдання впливатиме на тривалість її виконання). Наприклад, визначивши, що робітники – це керуючий ресурс, а бригадир – ні, можна домогтися скорочення термінів виконання завдання прокладки траншеї за рахунок призначення більшої кількості робочих. Збільшення ж кількості бригадирів не вплине на тривалість роботи [1].

При плануванні завантаження ресурсів може виникнути необхідність в описі нелінійного профілю споживання ресурсу окремим завданням. Project Planner дає можливість описати різні криві розподілу ресурсу, пропонуючи дев'ять стандартних кривих і можливість визначити власний профіль споживання, розбивши тимчасову фазу завдання на 10 періодів.

Засоби автоматичного перепланування завдань з урахуванням обмежень на ресурси набувають особливої важливості для великих проектів, коли менеджер не в змозі самотійно проаналізувати причини браку ресурсів і знайти рішення для кожної конкретної роботи. Project Planner дозволяє вибрати режим перерахунку розкладу і підібрати критерій перепланування робіт, що забезпечує отримання більш короткого розкладу. Серед режимів перерахунку можна виділити вирівнювання вперед (визначення можливої дати закінчення проекту при заданій початковій даті), вирівнювання назад (визначення самої пізньої допустимої дати початку проекту), згладжування перевантажень ресурсів в рамках тимчасових резервів робіт або в рамках заданого інтервалу [1].

Крім того, є можливість перерозподіляти призначення робіт між згрупованими ресурсами.

Засоби підтримки багатопроектного середовища управління в Project Planner включають можливість визначення ієрархії і права доступу до майстер-проекту і підпроектів. Менеджер-координатор проекту має право редагувати майстер проект і всі підпроекти. Менеджер підпроекту має право додавати ресурси в словник ресурсів, але не видаляти їх і не змінювати їх ціни. Якщо дозвіл ресурсних конфліктів в рамках підпроекту вимагає дані іншого підпроекту, менеджер може це зробити тільки при наділенні його додатковими повноваженнями з боку менеджера-координатора проекту. Однак, ресурсне планування по всьому проекту в цілому може здійснюватися тільки менеджером-координатором. Тільки він може визначити зв'язки між підпроектами. У порівнянні з багатьма іншими програмними продуктами, які також дають можливість багатопроектного управління, відмінною рисою P3 є докладний опис принципів багатопроектного управління в документації, де вони розглядаються з двох точок зору: менеджера-координатора проекту і менеджера підпроекту (хоча вважається, що тема мультипроектного управління вимагає додаткового підручника) [1].

Крім Project Planner, компанією Primavera Systems поставляється полегшена система для УП – SureTrak. Цей програмний продукт орієнтований на невеликі проекти, підпроекти, роботу конкретних виконавців з фрагментами проектів. SureTrak має ті ж засоби, в плані організації проекту за кодами і фільтрації інформації, установки обмежень і розрахунку розкладу, але в той же час існує ряд обмежень і додаткових можливостей.

З обмежень слід зазначити відсутність засобів багатопроектного управління і фрагментації проектів, меншу розмірність проектів, більш скромні засоби створення звітів. Однак в SureTrak з'явилися календарі ресурсів і, як наслідок можливість розрахунку тривалостей робіт з урахуванням узгодження календарів. Крім того, у ресурсів з'явилася додаткова категорія – дохід.

SureTrak здійснює импорт/експорт файлів у форматах Project Planner і MS Project. Таким чином, працюючи спільно, Project Planner і SureTrak пропонують масштабований підхід до управління проектами різного розміру і складності. Крім вищезазначених продуктів з сімейства Primavera інтерес може представляти система аналізу ризиків проекту Monte Carlo for Primavera [1].

Інша відома в світі управління проектами торгова марка – Artemis. Традиційно ПЗ сімейства Artemis використовувалося для управління великими інженерними проектами. На сьогоднішній день корпорація Artemis International поширює під цією торговою маркою серію програм під загальною назвою ArtemisViews.

Сімейство ArtemisViews складається з набору модулів, що автоматизують різні аспекти управління проектами: ProjectView, ResourceView, TrackView, CostView. Всі модулі сумісні за даними, працюють в архітектурі клієнт/сервер, підтримують ODBC стандарт і легко інтегруються з популярними СУБД Oracle, SQLBase, SQLServer, Sybase.

Кожен модуль може працювати як незалежно, так і в комбінації з іншим ПЗ. Ціна на це, традиційно недешеве ПЗ, розраховується виходячи з замовленої конфігурації.

ResourceView – спеціалізована система для планування і контролю використання ресурсів як в проектному або матричному середовищі управління, так і для поточних робіт. В системі реалізовані засоби підтримки узгодження керівниками розподілу ресурсів між роботами. Графічна панель управління ресурсами дозволяє менеджерам планувати, контролювати і оптимізувати їх завантаження за рахунок перерозподілу черги робіт відповідно до наявності ресурсів.

TrackView надає засоби ведення фактичної інформації по виконаним обсягам робіт, контролю за станом виконання і вартістю поточних робіт (проектних і позапроектних). Система дозволяє інтегрувати дані для різних рівнів управління в організації від рядових виконавців, які ведуть інформацію по своїм задачам, до вищого керівництва, яке може отримати узагальнені дані за фактичними витратами і обсягами робіт [1].

CostView забезпечує підтримку центрального сховища для інформації по всім витратам і доходам проектів. Пакет дозволяє аналізувати економічну ефективність контрактів, будувати таблиці грошових потоків, передбачати витрати і розраховувати показники внутрішньої норми рентабельності проектів. Безумовно ArtemisViews дозволяє створити потужне інтегроване рішення, однак, витрати, пов'язані з придбанням і впровадженням даного ПЗ, істотно обмежують коло потенційних користувачів.

Контрольні питання?

- 1) *Який існує взаємозв'язок між процесами управління?*
- 2) *Які групи процесів ви знаєте?*
- 3) *За якими критеріями можна визначити основні чи допоміжні процеси?*
- 4) *Назвіть особливості основних моделей управління?*
- 5) *В чому полягає суть ведення документації по проекту?*
- 6) *Як оцінити ризики проекту?*
- 7) *Які основні інструменти управління ризиками проекту?*
- 8) *Які відомі програмні засоби для управління проектами Ви знаєте?*

3 Моделювання бізнес-процесів

3.1 Функціональне моделювання SADT (IDEF0)

Більшість з тих, хто займається реалізацією проектів, пов'язаних зі створенням або розвитком корпоративних інформаційних систем, згодні з тезою, що замовнику потрібна інформаційна система, яка підвищує ефективність діяльності підприємства. Однак замовники і розробники інформаційних систем до сих пір розмовляють на різних мовах: вони по-різному розуміють – що значить підвищити ефективність підприємства.

Розробники інформаційних систем дуже часто під підвищенням ефективності розуміють зростання кількості робочих станцій у локальній обчислювальній мережі (ЛОМ) підприємства, зростання пропускної здатності ЛВС, зростання кількості документів, обробка яких здійснюється на автоматизованих робочих місцях (АРМах) і т.п.

Функціональна модель системи будується на основі функціональної діаграми. Крім чисто функціональних діаграм IDEF0 ця модель може включати діаграми, орієнтовані на дані, а саме DFD та IDEF3. Отже, до складу функціональної моделі можуть входити такі діаграми: • функціональні діаграми IDEF0; • діаграми потоків даних потоків даних DFD (Data Flow Diagramming); • діаграми опису послідовності процесів IDEF3 (Work Flow Diagramming); • діаграма дерева вузлів функціональної моделі (Node Tree Diagramming). У функціональній моделі діаграми IDEF0 відіграють головну роль. Діаграми DFD (потоків даних) і IDEF3 (опису послідовності процесів), як правило, доповнюють модель на нижніх рівнях декомпозиції, хоча вони можуть мати самостійне значення і будуватись як самостійні діаграми, починаючи з верхнього рівня. Діаграма Node Tree (дерева вузлів) просто демонстраційна, вона показує модель у загальному вигляді.

Для того, щоб замовник і розробник інформаційної системи розуміли один одного, необхідно, щоб розробник переорієнтувався з вирішення технічних завдань зі створення чи розвитку інформаційної системи на рішення комплексних завдань з підвищення ефективності діяльності підприємства замовника. При такому підході на перший план виступає проблема ефективного способу вивчення сфери діяльності замовника:

- обстеження існуючої бізнес-архітектури, ділових процесів, бізнес-правил, інформаційних потоків;
- ідентифікація проблем, “вузьких” місць, що негативно впливають на ефективність діяльності підприємства;
- розробка та реалізація заходів щодо усунення наявних проблем і зміни бізнес-архітектури підприємства, перебудові ділових процесів;
- розробка конкретного проекту корпоративної інформаційної системи, реалізація цього проекту та супровід в майбутньому.

В ході реалізації програми інтегрованої комп'ютеризації виробництва (ICAM), запропонованої свого часу ВПС для аерокосмічної промисловості США, була виявлена потреба в розробці методів аналізу взаємодії процесів у виробничих системах. Для задоволення цієї потреби була розроблена методологія IDEF0 (Integrated Definition Function Modeling), яка в даний час прийнята в якості федерального стандарту США. Методологія успішно застосовувалася в самих різних галузях, продемонструвавши себе як ефективний засіб аналізу, проектування та подання ділових процесів. В даний час методологія IDEF0 широко застосовується не тільки в США, але і у всьому світі. У Росії IDEF0 успішно застосовувався в державних установах (наприклад, у Державній Податковій Інспекції), в аерокосмічній промисловості (при проектуванні космодрому в Плесецьк), в Центральному Банку і комерційних банках Росії, на підприємствах нафтогазової промисловості та підприємствах інших галузей.

В основі IDEF0 методології лежить поняття блоку, який відображає деяку бізнес-функцію. Чотири сторони блоку мають різну роль: ліва сторона має значення “входу”, права – “виходу”, верхня – “управління”, нижня – “механізму” (рис. 3.1).

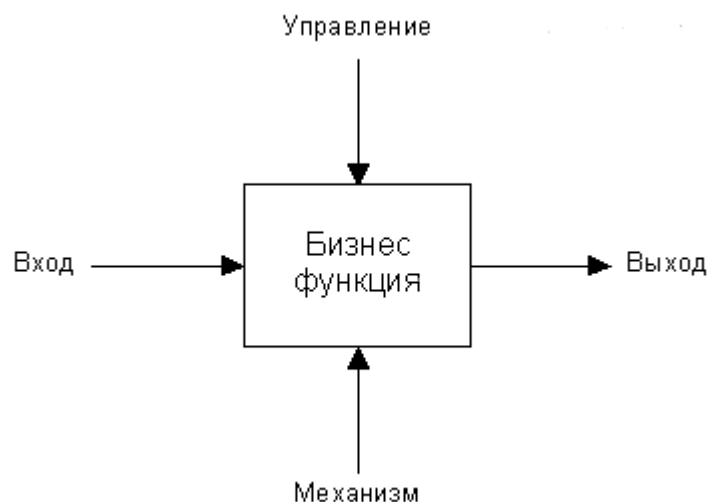


Рис. 3.1 – Загальний принцип методології IDEF0

Розглянемо загальні характеристики IDEF0 моделі. Дана модель, як уже відомо з попереднього, описує функції системи, тобто як система досягає своїх цілей, які процеси в ній відбуваються, як ці процеси пов’язані між собою. Модель являє собою деревовидну топологічну структуру і створюється на основі функціональної декомпозиції цілей та задач системи. Пояснимо, що це означає. Спочатку функції системи описуються в цілому без деталей. Цьому опису відповідає так звана контекстна діаграма. Вона має багато спільного з моделлю системи типу “Чорний ящик”, тільки в ній вказується не назва системи, а головна її функція (ціль, завдання). У подальшому контекстна діаграма піддається функціональній декомпозиції. У результаті останньої головна функція системи, описана в контекстній діаграмі, розбивається на підфункції (ціль - на підцілі,

завдання - на підзавдання). Потім кожна підфункція розбивається на більш дрібні підфункції і т. д. до досягнення найбільш простого ступеня деталізації. Результатом функціональної декомпозиції є розбиття функцій системи на елементарні процеси з точки зору аналізу, які можуть бути описані простими специфікаціями. Результат функціональної декомпозиції може бути зображений у вигляді ієрархічної моделі, яка має назву дерева вузлів функціональної моделі (Node Tree Diagramming).. Дерево вузлів є каркасом функціональної моделі, але не самою функціональною моделлю. Ця діаграма показує загальний склад моделі. Вигляд її зображений на рис. 3.2.



Рис. 3.2. - Приклад дерева вузлів функціональної моделі

Контекстна діаграма зображається прямокутником з вхідними й вихідними величинами. На відміну від моделі типу “Чорний ящик” у прямокутнику контекстної діаграми вказується не назва системи, а її основна функція (ціль системи). Входи й виходи контекстної діаграми розподілені не по двох, а по чотирьох сторонах прямокутника. Призначення цих сторін такі:

- Ліва сторона відповідає входам системи (input), величинам, які поступають у систему і переробляються нею у вихідні величини.
- Верхня сторона відповідає входам по керуванню (control), тобто різним керуючим діям, командам, стратегіям поведінки, процедурам, документам, що регламентують виконання роботи тощо. Ці величини не змінюються, а служать тільки для керування.
- Права сторона відповідає виходам системи (output), продуктам її діяльності, результатам перетворення вхідних величин, шкідливим виділенням, відходам тощо.

- Нижня сторона відповідає механізмам (mechanism), а саме засобам, ресурсам, за допомогою яких виконуються вказані в прямокутнику функції.
Вигляд контекстної діаграми показано на рис 3.3.

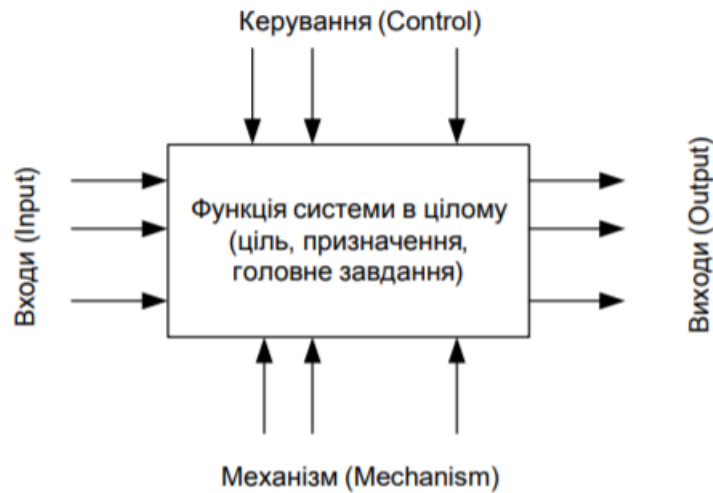


Рис. 3.3. – Загальний вигляд контекстної діаграми

У моделі IDEF0 існує п'ять типів зв'язків між функціональними блоками. Кожному з них відповідає певне розміщення дуг відносно блоків. Типи зв'язків такі:

- вхід – вихід,
- керування,
- вихід – механізм,
- зворотній зв'язок по керуванню,
- зворотній зв'язок по входу.

Вони зображені на рис. 3.4.

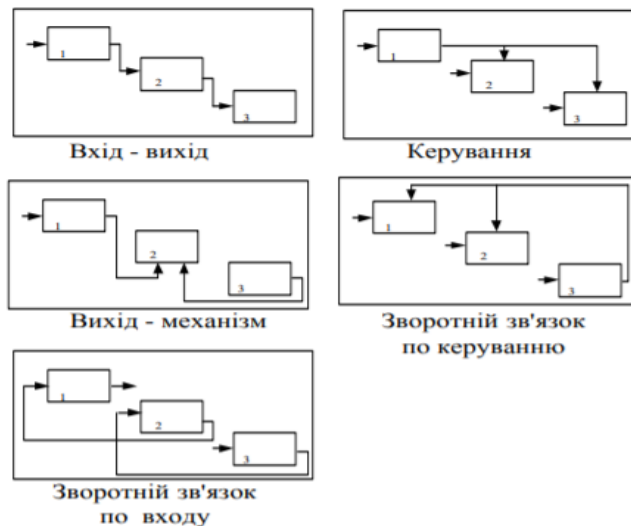


Рис. 3.4. - Типи зв'язків блоків функціональної моделі IDEF0

Фрагмент реальної моделі зі зворотнім зв'язком по керуванню показано на рис.3.5.

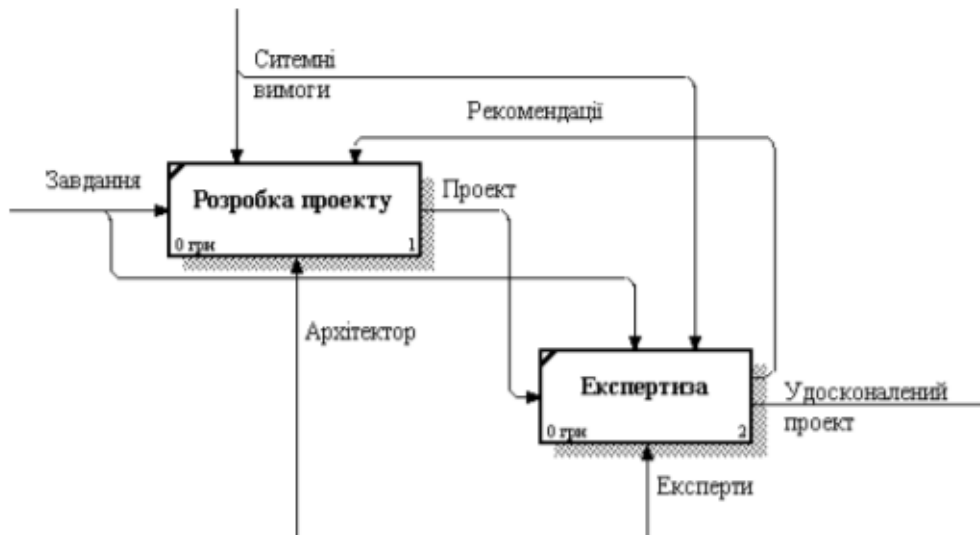


Рис. 3.5. - Зворотній зв'язок по керуванню

Функціональна модель будується на декількох сторінках і роз-руковується на аркушах паперу. На першому аркуші будують контекст-ну діаграму, на дру-гому аркуші - діаграму декомпозиції контекстної діаграми, на наступних арку-шах - діаграми декомпозиції кожного функ-ціального блоку. Таким чином, уся модель складається з ряду діаграм декомпозиції функціональних блоків. Крім діаграм декомпозиції до складу моделі може також входити діаграма вуз-лів функціональної мо-делі, окремі діаграми, які мають назву діаграм для ек-позиції (познача-ються FEO – діаграми), звіти по всій моделі, по стрілках, по блоках, по функціонально-вартісному аналізу та інші документи. Усі документи оформляють за певними правилами, нумерують, об'єднують в один до-кумент. Комп'ютерний варіант моделі міститься в пам'яті комп'ютера і може бути збе-режений у вигляді окремого файлу з розширенням *.bp1.

У IDEF0 реалізовані три базові принципи моделювання процесів:

- принцип функціональної декомпозиції;
- принцип обмеження складності;
- принцип контексту.

Принцип функціональної декомпозиції являє собою спосіб моделювання типової ситуації, коли будь-яка дія, операція, функція можуть бути розбиті (де-комповані) на більш прості дії, операції, функції. Іншими словами, складна бізнес-функція може бути представлена у вигляді сукупності елементарних фун-кцій. Представляючи функції графічно, у вигляді блоків, можна як би загляну-ти всередину блоку і детально розглянути її структуру і склад (див. рис. 3.6).

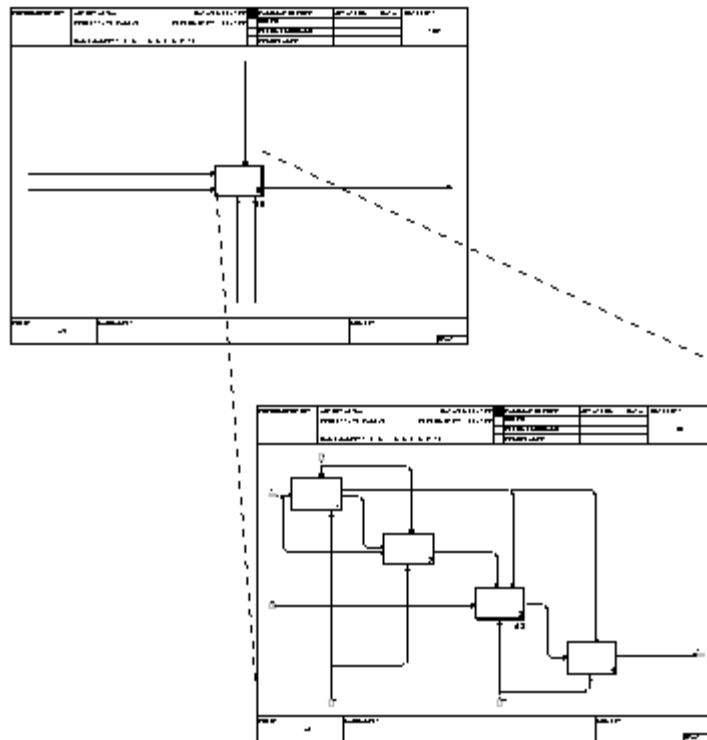


Рис. 3.6 – Графічне представлення структури в IDEF0

Після ознайомлення з базовими поняттями і принципами функціонального моделювання ділових процесів природним є питання: як це допомагає підвищувати ефективність і якість діяльності підприємства.

Побудова моделі ЯК Є. Обстеження підприємства є обов’язковою частиною будь-якого проекту створення чи розвитку корпоративної інформаційної системи. Побудова функціональної моделі ЯК Є дозволяє чітко зафіксувати, які ділові процеси здійснюються на підприємстві, які інформаційні об’єкти використовуються при виконанні ділових процесів і окремих операцій. Функціональна модель ЯК Є є відправною точкою для аналізу потреб підприємства, виявлення проблем і “вузьких” місць і розробки проекту вдосконалення ділових процесів.

Бізнес-правила. Модель ділових процесів дозволяє виявити і точно визначити бізнес-правила, які використовуються в діяльності підприємства.

На рис. 3 представлений фрагмент функціональної моделі документообігу. При виконанні операції “сортувати документи” використовується бізнес-правило: “реєстрації не підлягають: документи, надіслані в копії для зведення, телеграми та листи про дозвіл відряджень і відпусток ...”. Це правило зафіксовано в інструкції з документообігу. Функціональна модель дозволяє не тільки ідентифікувати існування цього правила, але також визначити, при виконанні якої операції і на якому робочому місці воно повинно застосовуватися.

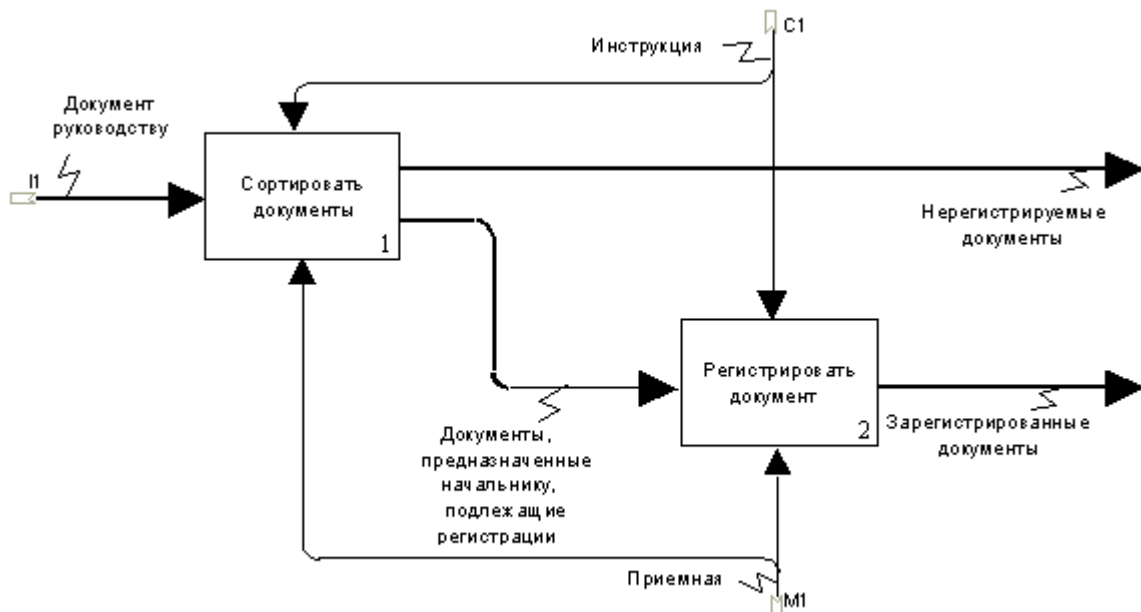


Рис. 3.7 – Приклад використання IDEF0

Дуже часто бізнес-правила на підприємстві не записані в інструкції: вони ніби є, але і їх нібито немає. В результаті спроби змінити що-небудь в діяльності підприємства або підрозділу можуть закінчитися невдачею тільки лише тому, що ці зміни суперечать сформованим бізнес-правилам.

Інформаційні об'єкти. Функціональна модель дозволяє ідентифікувати всі інформаційні об'єкти, якими оперує підприємство в своїй діяльності. На відміну від інформаційних моделей (Data Flow Diagrams, IDEF1X) функціональна модель IDEF0 відображає, як саме використовуються інформаційні об'єкти в рамках ділових процесів.

Побудова моделі ЯК БУДЕ. Створення та впровадження корпоративної інформаційної системи призводить до зміни умов виконання окремих операцій, структури ділових процесів і підприємства в цілому. Це призводить до необхідності зміни системи бізнес-правил, використовуваних на підприємстві, модифікації посадових інструкцій співробітників. Функціональна модель ЯК БУДЕ дозволяє вже на стадії проектування майбутньої інформаційної системи визначити ці зміни. Застосування функціональної моделі ЯК БУДЕ дозволяє не тільки скоротити терміни впровадження інформаційної системи, але також знизити ризики, пов'язані з несприйнятливістю персоналу до інформаційних технологій.

Розподіл ресурсів. Функціональна модель дозволяє чітко визначити розподіл ресурсів між операціями ділового процесу, що дає можливість оцінити ефективність використання ресурсів.

Особливо ця задача актуальна при створенні нових ділових процесів на підприємстві. Наприклад, компанія, яка спеціалізується на розробці замовленого програмного забезпечення, прийняла рішення створити власну службу збуту. Функціональна модель ділового процесу з продажу програмного забезпечення дозволить керівництву компанії чітко визначити, які ресурси необхідно виділи-

ти для того, щоб забезпечити функціонування служби збуту, скільки співробітників необхідно залучити для роботи в новій службі, які функціональні обов'язки ці співробітники повинні виконувати і т.д.

4. Побудова функціональної моделі системи (моделі IDEF0)

Перед визначенням і описом порядку побудови моделі слід зробити декілька зауважень:

1. Всяка SADT модель вимагає точного визначення границь системи, цілей моделювання, точки зору, контексту розгляду системи.
2. SADT моделі вимагають, щоб система розглядалась весь час з однієї точки зору, оскільки зміна точки зору може зробити модель неадекватною. Точка зору диктує підбір потрібної інформації та спосіб її подачі.
3. SADT моделі будуються з верхнього рівня "з голови". У них діаграми нижчого рівня є деталізованими діаграмами верхнього рівня. Кінцевий результат це ієрархічна структура діаграм.

Кінцевий результат це ієрархічна структура діаграм. На початку побудови функціональної моделі необхідно чітко визначити:

- границі системи,
- ціль моделювання,
- контекст розгляду системи
- точку зору,

тобто відповісти на запитання, які є вихідними усякого системного аналізу.

Побудова функціональної моделі полягає у функціональній декомпозиції системи. Виконуючи декомпозицію, окремі функції системи розбивають на підфункції і зображують у вигляді ієрархічної структури. Під час аналізу систем, які існують, а не проектуються, найбільш часто функціональну декомпозицію виконують на основі реальної структурної схеми системи. У першу чергу визначають функції окремих підсистем і ставлять їм у відповідність функціональні блоки. Функціональні блоки не обов'язково повинні відповідати структурним одиницям системи, оскільки кожна структурна одиниця може виконувати декілька

функцій або одну функцію можуть здійснювати декілька підсистем. Допускається виконувати функціональну декомпозицію незалежно від структурної схеми системи, а орієнтуватись тільки на функції, які повинні виконувати кожна підсистема. Це часто роблять з ціллю реінженірингу систем, тобто з ціллю перебудови і удосконалення роботи системи. Для систем, які проектуються, функціональну декомпозицію виконують раніше розробки структурної схеми, а структурну схему розробляють вже на основі функціональної моделі

Функціональна модель являє собою ряд діаграм, а саме контекстну діаграму, діаграми декомпозиції та деяких інших документів. Елементи контекстної діаграми і діаграм декомпозиції є такими:

- функціональні блоки (роботи) – activity;
- інтерфейсні дуги (стрілки) - arrow.

Функціональні блоки діаграм – роботи (activity) зображаються прямокутниками. Сторони прямокутника мають таке призначення:

- вхід (input);
- керування (control);
- вихід (output);
- механізм (mechanism).

Кожен блок повинен мати принаймні одну керуючу і одну вихідну величини. Зрозуміло, що блок, який не має вихідної величини, на діаграмі не повинен бути (це робота, яка не дає ніякого результату). Відсутність вказаних дуг при синтаксичному аналізі діаграми буде вважатися помилкою. Інтерфейсні дуги (стрілки) – arrow на діаграмах декомпозиції встановлюють зв'язки між окремими функціональними блоками (роботами). Слово “інтерфейс” означає зв'язок, англійською мовою - це те, що стоїть між обличчями. Тобто дуги зв'язують роботи між собою. Вони відповідають тим об'єктам, які циркулюють між блоками, передаються від блоку до блоку при функціонуванні системи. Усі дуги повинні мати імена. Під час присвоєння імені дузі вказують опис об'єктів, яким вона відповідає. Імена дугам присвоюють у місцях встановлених міток. Мітки на дугах слід розставляти відповідно до таких правил:

- Кожна дуга повинна мати ім'я.
- Після розгалуження дузі можна присвоювати нове ім'я або не присвоювати.
- Якщо після розгалуження дузі не присвоєно нове ім'я, то вона містить всі об'єкти, які містила до розгалуження.
- Якщо після розгалуження дуга містить частину об'єктів, то вона має бути поійменована і повинен бути вказаний (чи зрозумілий з назви) перелік об'єктів, яким вона відповідає.
- Після злиття дуга обов'язково повинна бути поійменована.

На діаграмах декомпозиції допускається тунелювання дуг. Тунелювання - це умовне позначення дуг без безпосереднього нанесення їх на діаграму. Тунелювання має на меті спростити зображення діаграм і облегшити їх читання. Дійсно, якщо діаграма декомпозиції має, наприклад, 6 функціональних блоків і кожен блок має по декілька дуг з різних сторін, то ці дуги можуть так переплітатись, що читати діаграму буде важко. Тунелювання дозволяється виконувати тільки для другорядних дуг. Дуги, що виходять з батьківської діаграми і відносяться до усіх блоків дочірньої діаграми, можуть не зображатися. Закінчення такої дуги біля виходу з батьківської діаграми береться в дужки (тунелюється) і в подальших діаграмах не вказується. Таке позначення називають “не в дочірній діаграмі”. Тунелювання може бути і таке, що дуга не вказується у батьківській діаграмі, а з'являється в дочірній. Така дуга позначається дужками біля входу у блок, що означає “не на батьківській діаграмі”. Розглянемо порядок побудови функціональної моделі. Використовуючи вказані правила, функціональну модель можна побудувати як безпосередньо на папері за допомогою олівця, так і за допомогою комп'ютера з використанням пакету програм VPwin. Побудова діаграм на папері порівняльно легка, коли вся

модель має один, два рівні деком-позиції. Якщо таких рівнів більше, то виникають труднощі, які доволі важко перебороти. Тому функціональну модель рекомендується будувати за допомогою комп'ютера. Правила роботи з комп'ютером і введення моделі вказані у [45] і тут їх детально розглядати не будемо, а тільки відзначимо головні моменти побудови. Зауважимо, що перед початком роботи треба забезпечити на носії інформації, а саме на вінчестері комп'ютера місце для розміщення моделі, тобто створити папку і в подальшому розміщувати всі результати роботи саме в цій папці.

Після завантаження програмного пакету VPwin необхідно:

- відкрити новий документ;
- вибрати тип моделі IDEF0;
- присвоїти їй власне ім'я;
- вказати відомості про автора розробки;
- зберегти файл моделі у попередньо створеній папці. Після цього приступають до введення інформації за створюваною моделлю. Ця інформація повинна містити:

- визначення функцій системи;
- цілі моделювання;
- контекст розгляду системи;
- точку зору, з якої розглядується система;
- відомості про авторів розробки;
- перелік джерел інформації.

Після побудови контекстної діаграми переходять до побудови наступних діаграм моделі. Ці діаграми є діаграмами декомпозиції. Діаграму декомпозиції будують у такому порядку. Вибравши значок декомпозиції, в діалоговому вікні вказують тип діаграми, яка буде побудована. (Пакет VPwin дозволяє будувати складні моделі з різними типами діаграм.) Треба вказати діаграму IDEF0 і кількість блоків декомпозиції. Діалогове вікно вибору кількості блоків декомпозиції за типом діаграми декомпозиції показано на рис. 3.8.



Рис. 3.8 - Діалогове вікно вибору типу діаграми і кількості блоків деком-позиції

Кількість блоків декомпозиції рекомендується вибирати в границях від 2 до 6. Обмеження зумовлені тим, що не має сенсу декомпозиції, яка замінює один блок іншим, а число 6 рекомендоване тому, що діаграма, в якій більше 6 блоків і зв'язків між ними, стає складною і читати її важко (в пакеті VPwin допускається вводити до 8 блоків). Після вибору типу діаграм та кількості

блоків робіт з'являється бланк діаграми декомпозиції на зразок, показаний на рис. 3.9. Бланк діаграми декомпозиції містить блоки робіт, розміщені зверху – вниз, зліва – направо. Використовуючи цей зразок, дамо пояснення змісту діаграм декомпозиції та правил їх побудови. Розміщення функціональних блоків зверху – вниз, зліва – направо називають домінантним. У верхньому лівому куті розміщується блок роботи, яку вважають найбільш пріоритетною, нижче і правіше знаходяться блоки менш пріоритетних функцій, або у випадку, коли діаграма зображує поопераційне виконання певних робіт, то домінантність блоків визначає порядок виконання робіт, яким ці блоки відповідають. Блоки функціональної діаграми мають порядковий номер і закреслений лінією верхній кут. Нумерація блоків відповідає їх пріоритетності й служить для встановлення єдиного порядку блоків у всій моделі. Закреслений лінією лівий верхній кут блоку є ознакою, що даний блок ще не піддавався операції декомпозиції. Якщо у подальшому виконати декомпозицію цього блоку, то помітка буде знята, якщо не виконувати, то вона залишиться на діаграмі.

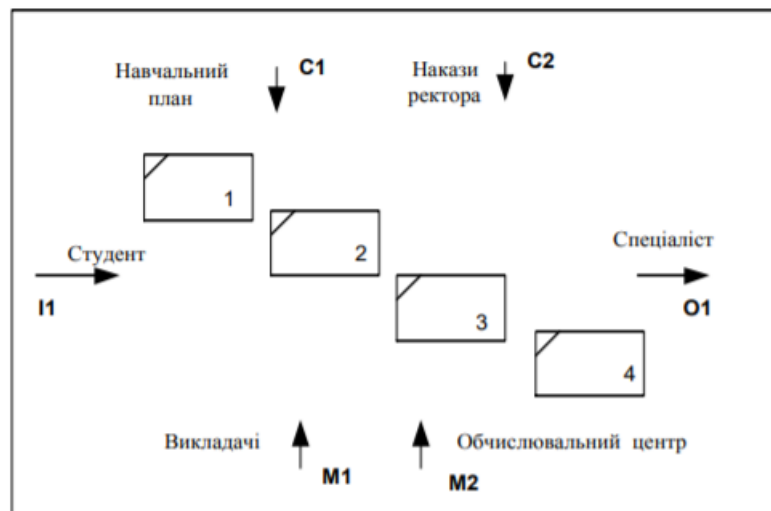


Рис. 3.9 - Зразок бланку декомпозиції

Після побудови діаграми декомпозиції першого рівня приступають до декомпозиції її функціональних блоків. Декомпозицію виконують аналогічно декомпозиції контекстної діаграми з тією різницею, що перед декомпозицією потрібно вказати, який саме блок діаграми піддається декомпозиції. Виконавши декомпозицію діаграми першого рівня, необхідно виконати аналіз моделі на наявність синтаксичних помилок, виконати експертизу на адекватність моделі, а потім приступити до декомпозиції діаграм другого рівня. Аналогічно поступають і після декомпозиції діаграм другого і наступних рівнів. Закінчення декомпозиції визначають за вимогами цілі моделювання, та за тією ознакою, що блоки декомпозиції є елементарними роботами. Декомпозиції можна піддавати не всі роботи, а тільки ті, які, виходячи з цілей моделювання, вимагають виконання декомпозиції. По завершенні декомпозиції моделі одержується цілий ряд діаграм на зразок, показаний на рис. 3.10.

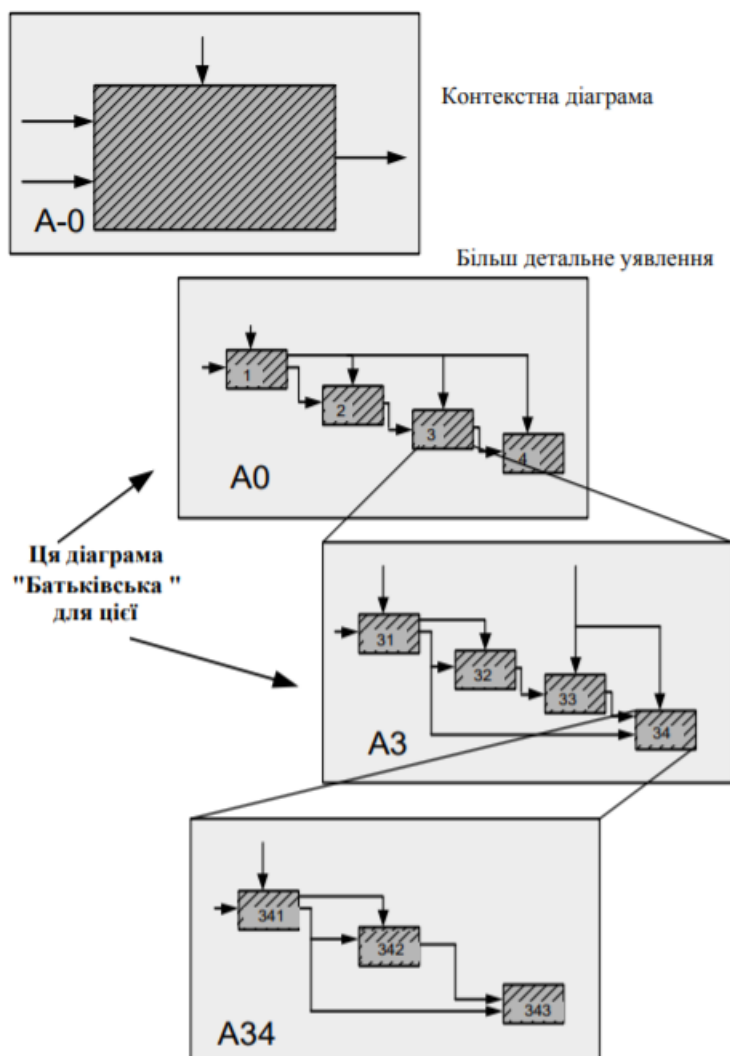


Рис. 3.10 - Розміщення діаграм декомпозиції функціональної моделі

Кожна діаграма функціональної моделі системи має позначення. Система позначення є такою. Контекстна діаграма позначається А - 0. Декомпозиція контекстної діаграми (діаграма першого рівня декомпозиції) - А0, декомпозиція функціональних блоків діаграми першого рівня - А1, А2, А3, ..., декомпозиція блоків діаграми А1 - двома цифрами А11, А12, А13, ..., блоків діаграми А2 - двома цифрами А21, А22, А23, Наступні діаграми декомпозиції позначаються літерою А з трьома цифрами, подальші (якщо вони є) з чотирма цифрами і т. д. Проте більшість моделей обмежується діаграмами 3 – 4 рівня, оскільки дерево декомпозиції стає великим у глибину і модель є складною. Для великих організаційних систем модель може включати сотні, або тисячі аркушів. у разі необхідності працювати з моделями, що мають багато діаграм на різних рівнях декомпозиції, їх розбивають на ряд менш складних моделей і розглядають самостійно. Таке розбиття і з'єднання декількох моделей в одну передбачене пакетом ВРwin. Воно застосовується, коли виникає необхідність колективу аналітиків працювати з великими системами. Це

дозволяє кожному виконувати свою роботу незалежно від інших, а потім об'єднувати результати роботи. Приклад функціональної моделі для нескладної системи “Токар- на дільниця” наведено на рис. 3.11, 3.12.



Рис. 3.11. - Контекстна діаграма системи “Токарна дільниця”

На рис. 3.11, 3.12 показаний приклад побудови функціональної моделі системи “Токарна дільниця”. Модель складається з двох діаграм: контекстної і діаграми декомпозиції.



Рис. 3.12. - Діаграми декомпозиції системи “Токарна дільниця”

Діаграма декомпозиції має у своєму складі три роботи, а саме: “Обточка деталі”, “Перевірка якості деталі” та “Аналіз браку”. Пріоритетною і першою за чергою є робота “Обточка деталі”. Токарі відповідно до креслень обточують на верстатах заготовки і одержують деталь. Виготовлені деталі передають на перевірку якості. Перевірка здійснюється контролерами відповідно до креслень. Перевірені деталі, залежно від результату перевірки, передають в інші системи (є вихідними величинами), або бракуються і передаються для аналізу причин браку. Після аналізу причин браку, який здійснюють контролери, деякі деталі знову надходять на обточку, а інші бракуються повністю і виходять із системи як невиправний брак. Після аналізу причин браку інформація про результати аналізу передається токарям, які обточують деталі. Ця інформація

служить для них керуючою інформацією і призначена для покращання роботи.

3.2 Моделювання процесів (IDEF3)

Основу діяльності будь-якої організації становлять її ділові процеси, або бізнес-процеси, які визначаються цілями й завданнями діяльності суб'єкта господарювання. Процеси забезпечують реалізацію всіх видів діяльності підприємства, пов'язаних з виробництвом товарів і/або послуг. Для кожного виду роботи, що входить у загальний процес господарської діяльності, визначені тимчасові характеристики, які визначають її місце в загальній послідовності робіт, умови ініціації й час виконання

Використовується кілька різних методів, основою яких є як структурний, так і об'єктно-орієнтовний підходи до моделювання. До числа найпоширеніших методів відносяться: метод функціонального моделювання SADT (IDEF0); метод моделювання процесів IDEF3; моделювання потоків даних DFD; метод ARIS; метод Ericsson-Penker [2].

На сьогодні найвідомішими мовами (нотаціями) графічного моделювання бізнес-процесів є UML, ARIS, IDEF (IDEF0, IDEF3 у програмній інтерпретації BPwin), BPMN.

В даний час на ринку комп'ютерних технологій для моделювання бізнес-процесів існує достатня кількість спеціальних програм, що дозволяють обстежувати підприємство і будувати моделі. Це такі програмні засоби, як Ramus, Business Studio, Fox Manager, BPwin та ін. [4]. Аналіз та моделювання бізнес-процесів були проведені на базі товариства ПАТ «АЗОТ», зокрема, досліджена діяльність одного з його цехів – А-5 з виробництва аміаку. Для відображення функціонування діяльності використано програмний продукт Ramus – крос-платформна система моделювання та аналізу бізнес-процесів. В роботі допомогою програми Ramus побудовано загальну контекстну діаграму діяльності цеху А-5, яка зображена на рисунку 3.13.

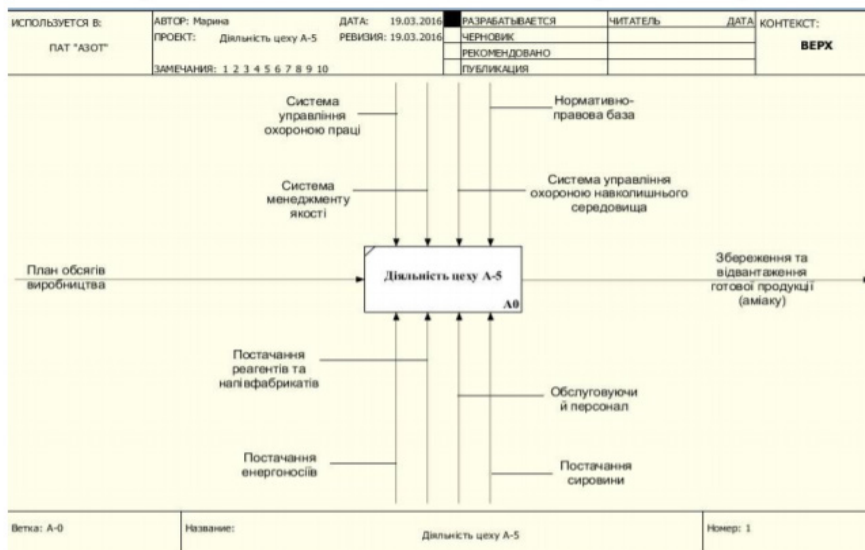


Рис. 3.13 – Контекстна діаграма діяльності цеху А-5 за методологією IDEF0

З рисунку 1 видно, що діяльність цеху можна представити контекстною діаграмою, на якій відображено вхідна інформація (бізнес-процеси), механізми, керуючого впливу та вихідна інформація. На вхід контекстної діаграми моделювання бізнес-процесів цеху А-5 подаються планові завдання (обсяги виробництва), які встановлюється для цеху керівництвом підприємства. До механізмів впливу і ресурсів відносяться: постачання сировини, реагентів, напівфабрикатів та енергоносіїв необхідних для виробництва аміаку; обслуговуючий персонал. Керуючий вплив містить у собі; нормативно-правову базу; систему менеджменту якості; систему управління охороною праці та навколишнього середовища. На виході моделі відображені бізнес-процеси зберігання та відвантаження готової продукції (аміаку).

На рисунку 2 зображений деталізований процес виробництва аміаку у вигляді моделі бізнес-процесів за методологією IDEF3. З рисунку 2 видно, що модель бізнес-процесів за методологією IDEF3, яка описує етапи технологічного циклу виробництва аміаку, складається з декількох етапів. Першим етапом є отримання плану виробництва. Наступним – підготовка до виробництва, що включає постачання матеріалів та ресурсів. Третій етап – це безпосередньо технологічний процес виробництва аміаку, який може протікати по різному: під час виробництва можуть виникати проблеми технічного характеру, в такому випадку відбувається ремонт пошкоджень і підприємство знову повертається до другого етапу; проблеми якості продукції – відбувається аналіз методів виробництва та їх зміна, і в такому випадку технологічний цикл знову повертається до другого етапу; якщо технологічний цикл виробництва аміаку проходить без збоїв, то технологічний цикл завершується четвертим етапом – зберігання або відвантаження аміаку.

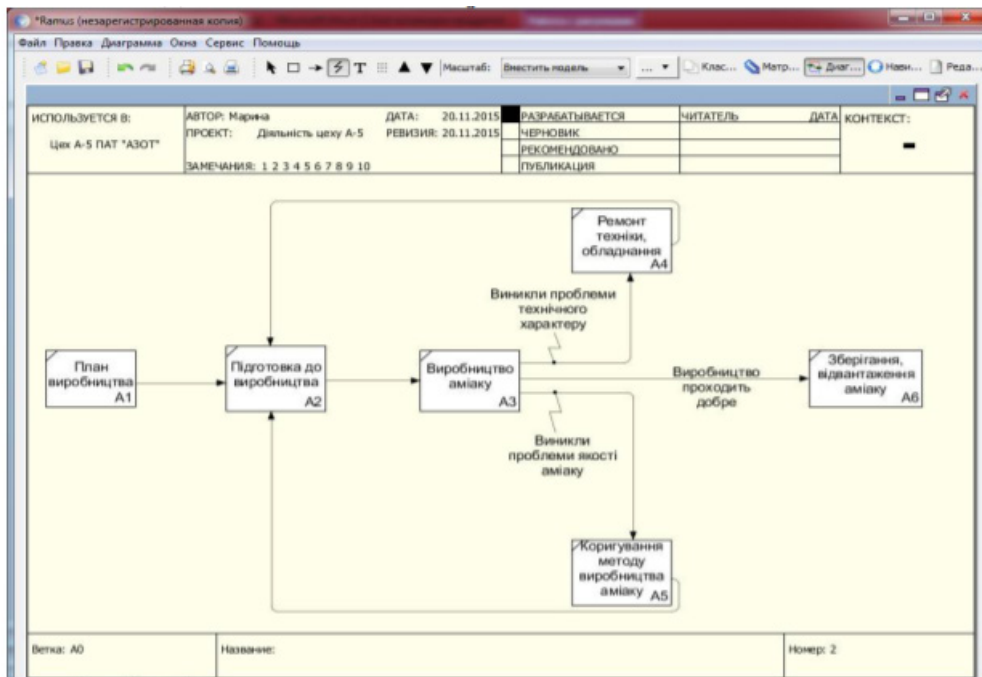


Рис.3.14 - Моделі бізнес-процесів виробництва аміаку за методологією IDEF3

Отже, моделювання бізнес процесів для цеху А-5 дає можливість пошуку шляхів підвищення якості та швидкості виробництва продукції з одночасним зниженням витрат, зростанням професіоналізму співробітників; підвищенням конкурентоспроможності продукції.

Під моделлю бізнес-процесу будемо розуміти його формалізований (графічний, табличний, текстовий, символічний) опис, що відображає реально існуючу або ймовірну діяльність організації. Модель бізнес-процесу, як правило, містить такі дані:

- набір кроків процесу
- бізнес-функцій;
- порядок виконання бізнес-функцій;
- механізми контролю і управління в рамках бізнес-процесу;
- виконавців кожної бізнес-функції;
- вхідні документи/інформацію, вихідні документи/інформацію;
- ресурси, необхідні для виконання кожної бізнес-функції;
- документацію/умови, що регламентують виконання кожної бізнес-функції;
- параметри, які характеризують виконання бізнес-функцій і процесу в цілому.

Методи моделювання бізнес-процесів можна умовно поділити за двома основними принципами: структурні методи та об'єктно-орієнтовані. На рис. 3.15 наведено класифікацію методів моделювання бізнес-процесів.

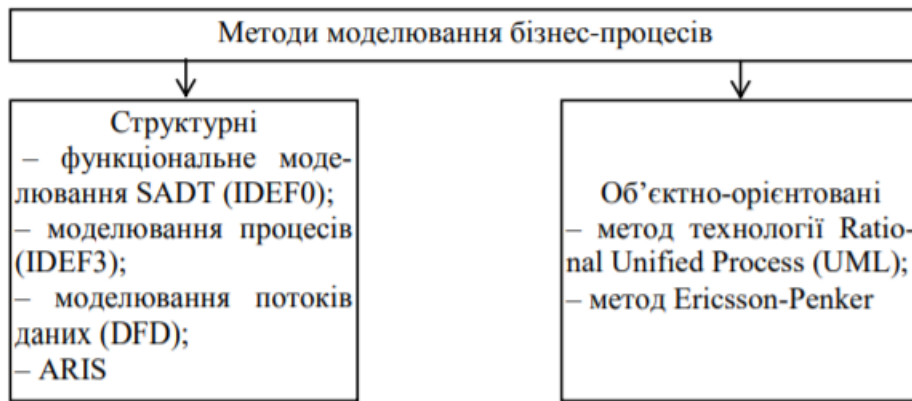


Рис. 3.15 – Класифікація методів моделювання бізнес-процесів

Метод SADT (Structured Analysis and Design Technique) є методом процесного підходу до управління і може використовуватись для моделювання різноманітних бізнес-процесів і систем.

Метод IDEF3 призначений для моделювання послідовності виконання дій і взаємозалежностей між ними в рамках процесів.

Метод моделювання потоків даних DFD (Data Flow Diagrams) дозволяє продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Методика моделювання ARIS (Architecture of Integrated Information System) ґрунтується на теорії побудови інтегрованих інформаційних систем. ARIS підтримує чотири типи моделей:

- організаційні;
- функціональні;
- інформаційні;
- моделі керування.

З наведеного вище аналізу методів моделювання бізнес-процесів авторами пропонується використовувати методологію IDEF0 та IDEF3 для побудови концептуальних моделей бізнес-процесів функціонування управляючих веб-порталів. За допомогою IDEF0 будуватиметься модель на найбільш абстрактному рівні, подальша деталізація проводитиметься у вигляді IDEF3-діаграм.

Розглянемо побудову концептуальної моделі на прикладі бізнес-процесу «Ліцензування діяльності з надання освітніх послуг у сфері вищої освіти». На рис. 4 представлена IDEF0-діаграма.

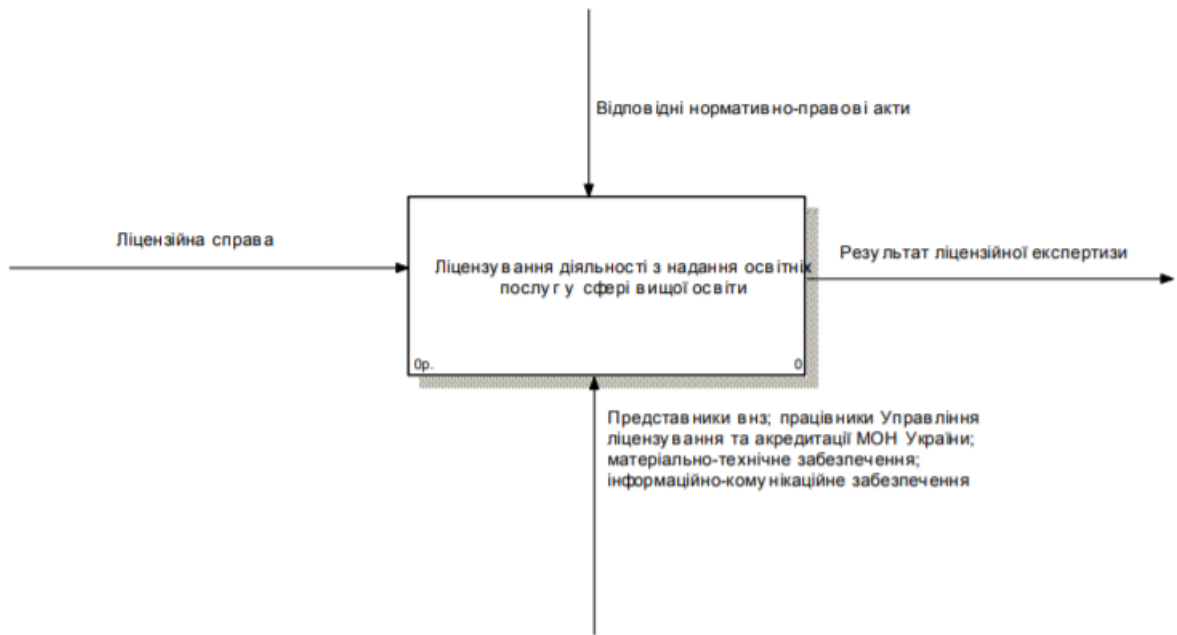


Рис. 3.16 – Функціональна діаграма бізнес-процесу «Ліцензування діяльності з надання освітніх послуг у сфері вищої освіти»

Деталізація функціонального блоку представлена у вигляді діаграми IDEF3 на рис. 3.17.

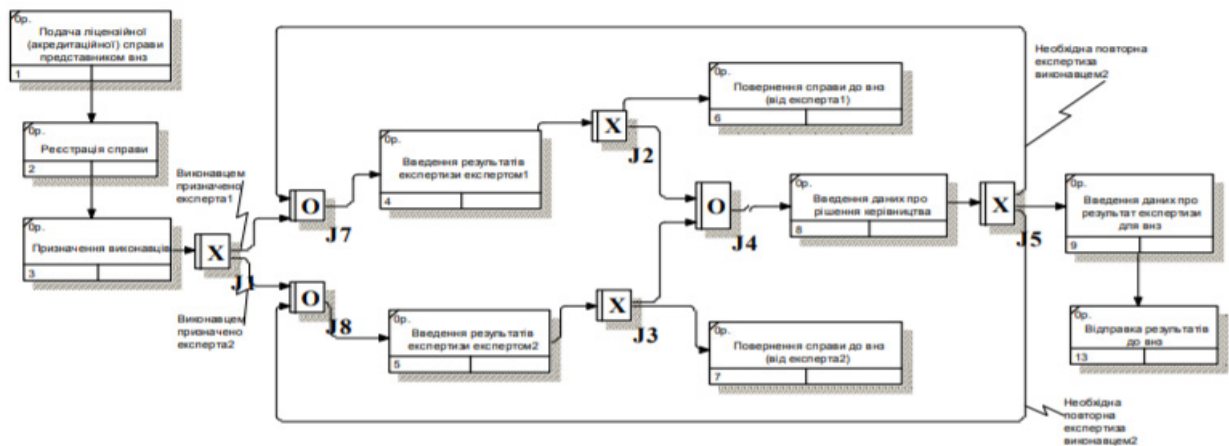


Рис. 3.17 - IDEF3-діаграма бізнес-процесу «Ліцензування діяльності з надання освітніх послуг у сфері вищої освіти»

Як видно з рис. 3.2.5, діаграма IDEF3 містить набір бізнес-функцій (дій або етапів виконання бізнес-процесу), зв'язків, посилань і перехресть [2]. Перехрестя використовуються для відображення безлічі дій, які можуть або повинні бути завершені перед початком наступної дії. Розрізняють перехрестя для злиття (Fan-in Junction) і розгалуження (Fan-out Junction) стрілок. Перехрестя не може використовуватись одночасно для злиття й для розгалуження. Існують декілька типів перехресть, зміст кожного з них наведений у табл. 3.1.

Таблиця 3.1. Типи перехресть

Найменування	Зміст у випадку злиття стрілок (Fan-in Junction)	Зміст у випадку розгалуження стрілок (Fan-out Junction)
Asynchronous AND	Всі попередні процеси повинні бути завершені	Всі наступні процеси повинні бути запусчені
Synchronous AND	Всі попередні процеси завершені одночасно	Всі наступні процеси запускаються одночасно
Asynchronous OR	Один або кілька попередніх процесів повинні бути завершені	Один або кілька наступних процесів повинні бути запусчені
Synchronous OR	Один або кілька попередніх процесів завершені одночасно	Один або кілька наступних процесів запускаються одночасно
XOR (Exclusive OR)	Тільки один попередній процес завершений	Тільки один наступний процес запускається

Якщо в рамках кожної дії бізнес-процесу на діаграмі IDEF3 виконуються інші дії, потрібно виконувати їх подальшу декомпозицію у вигляді дочірніх IDEF3-діаграм. Для наведеного бізнес-процесу подальша декомпозиція не потрібна.

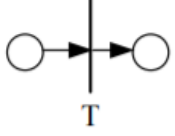
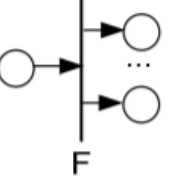
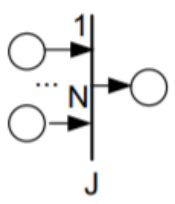
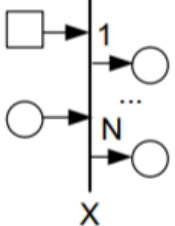
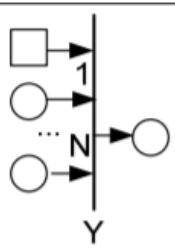
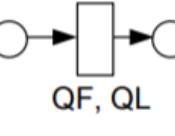
Формалізований опис бізнес-процесів

Концептуальні моделі дозволяють якнайкраще представити логіку виконання бізнес-процесу, але їх не можна застосовувати для управління процесами. У зв'язку з цим виникає необхідність перетворення концептуальних моделей у формалізований опис, який можна використовувати в подальшому для управління роботою web-порталу

В управляючих Е-мережах базисний набір переходів звичайних Е-мереж розширений шляхом введення переходів-черг, що виконують функції розміщення міток у черзі з різною дисципліною обслуговування.

Опис схем спрацьовування всіх типів переходів SEN приведений в табл. 3.2.

Таблиця 3.2. Базисний набір переходів CEN

Граф переходу	Умова $\sigma(t)=1$	Схема спрацьовування
 T	$(M(x) = 1 \wedge (M(y) = 0$ $x \wedge \bullet t, y \wedge t \bullet$	$(M'(x) = 0);$ $(M'(y) = 1);$ $z_0 : d_{xy} = d_{xy}$
 F	$(M(x) = 1) \wedge$ $\forall y \in t \bullet : (M(y) = 0)$	$(M'(x) = 0);$ $\forall y \in t \bullet : (M'(y) = 1);$ $z_0 : \forall y \in t \bullet : d_{yj} = d_{xy}$
 J	$\forall x \in \bullet t : (M(x) = 1) \wedge$ $(M(y) = 0)$	$\forall x \in \bullet t : (M'(x) = 0);$ $(M'(y) = 1);$ $r_0 : d_{yj} = d_{x1,j}$
 X	$(M(x) = 1) \wedge (M(y_r) = 0)$	$(M'(x) = 0);$ $(M'(y_r) = 1);$ $z_0 : d_{y_r,j} = d_{xy}$
 Y	$(M(x_r) = 1) \wedge (M(y) = 0)$	$(M'(x_r) = 0);$ $(M'(y) = 1);$ $z_0 : d_{yj} = d_{x_r,j}$
 QF, QL	1) $(M(x) = 1) \wedge (M(y) = 0)$ 2) $(M(x) = 1) \wedge (M(y) = 1)$	1) $M'(x) = 0; M'(y) = 1$ 2) $M'(x) = 0; M'(y) = 0;$ $M'(y) = 1$ $z_0 : d_{yj} = d_{yj}$

Передбачено чотири різновиди переходів-черг, які реалізуються в рамках двох ос- новних типів цих переходів:

- перехід QF реалізує порядок обслуговування міток FIFO і пріоритетне обслугову- вання за збільшенням пріоритетів;
- перехід-черга QL реалізує дисципліну обслуговування LIFO і пріоритетне обслу- говування за зменшенням пріоритетів.

Якщо під час переміщення мітки в чергу вихідна позиція вже зайнята однією з тих міток, що знаходяться в черзі, то правило переміщення міток буде таким:

- вилучена з вхідної позиції x мітка поміщається в чергу;

– вилучена з вихідної позиції у мітка поміщається в чергу, звільняючи позицію;

– у вихідну позицію з черги поміщається мітка, вибрана відповідно до прийнятої дисципліни обслуговування черги.

Слід вказати на обмеження, пов'язані з використанням переходів-черг.

1. Процедура перетворення виконується під час переміщення мітки у вхідну позицію.

2. Функція тимчасової затримки для них не визначається.

3. Для правильної роботи даних переходів в одній структурній зв'язці з їх вихідними позиціями повинні знаходитися переходи з нульовим часом затримки. Інакше можуть бути порушені правила спрацьовування переходів.

3.3 Моделювання потоків даних (DFD)

Діаграма потоків даних ([англ.](#) *Data Flow Diagram*) — модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма потоків даних також може використовуватись для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму потоків даних рівня контексту, завдяки чому буде показано взаємодію системи із зовнішніми модулями. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати розлого розроблювану систему.

Діаграми потоків даних містять чотири типи графічних елементів:

- процеси - являють собою трансформацію даних в рамках описуваної системи;
- сховища даних (репозиторії);
- зовнішні по відношенню до системи сутності;
- потоки даних між елементами трьох попередніх типів

Інформаційна система приймає ззовні потоки даних. Для позначення елементів середовища функціонування системи використовується поняття зовнішньої сутності. Усередині системи існують процеси перетворення інформації, що породжують нові потоки даних. Потоки даних можуть надходити на вхід до інших процесів, міститися (і витягатися) у накопичувачі даних, передаватися до зовнішніх сутностей.

Модель DFD, як і більшість інших структурних моделей - ієрархічна модель. Кожен процес може бути підданий декомпозиції, тобто розбиті на структурні складові, відносини між якими в тій же нотації можуть бути показані на окремій діаграмі. Коли досягнута потрібна глибина декомпозиції - процес нижнього рівня супроводжується міні-специфікацією (текстовим описом).

Метою методики потоків даних є побудова моделі розглянутої системи у вигляді діаграми потоків даних (Data Flow Diagram - DFD), що забезпечує правильний опис виходів (відгуків системи у вигляді даних) при заданому впливі на вхід системи (подачі сигналів через зовнішні інтерфейси). Діаграми потоків даних є основним засобом моделювання функціональних вимог до проекрованої системи.

При створенні діаграми потоків даних використовуються чотири основних поняття:

- потоки даних,

Потоки даних є абстракціями, що використовуються для моделювання передачі інформації (або фізичних компонентів) з однієї частини системи в іншу. Потоки на діаграмах зображуються іменованими стрілками, орієнтація яких указує напрямок руху інформації.

- процеси (роботи) перетворення вхідних потоків даних у вихідні,

Призначення процесу (роботи) складається в продукуванні вихідних потоків із вхідних відповідно до дії, що задає ім'ям процесу. Ім'я процесу повинне містити дієслово в невизначеній формі з наступним доповненням (наприклад, "одержати документи по відвантаженню продукції"). Кожен процес має унікальний номер для посилань на нього усередині діаграми, що може використатися разом з номером діаграми для одержання унікального індексу процесу у всій моделі.

В основі цієї методології лежить побудова моделі аналізованої ІС - проекрованої або реально існуючої. Відповідно до методології модель системи визначається як ієрархія діаграм потоків даних (ДПД або DFD), що описують асинхронний процес перетворення інформації від її введення у систему до видачі користувачу. Діаграми верхніх рівнів ієрархії (контекстні діаграми) визначають основні процеси або підсистеми ІС із зовнішніми входами і виходами. Вони деталізуються за допомогою діаграм нижнього рівня. Така декомпозиція продовжується, створюючи багаторівневу ієрархію діаграм, доти, поки не буде досягнутий такий рівень декомпозиції, на якому процеси стають елементарними і деталізувати їх далі неможливо.

Джерела інформації (зовнішні сутності) породжують інформаційні потоки (потоки даних), що переносять інформацію до підсистем або процесів. Ті у свою чергу перетворюють інформацію і породжують нові потоки, які переносять інформацію до інших процесів або підсистем, нагромаджувачів даних або зовнішніх сутностей - споживачам інформації. Таким чином, основними компонентами діаграм потоків даних є:

- зовнішні сутності;
- системи/підсистеми;
- процеси;
- нагромаджувачі даних;
- потоки даних.

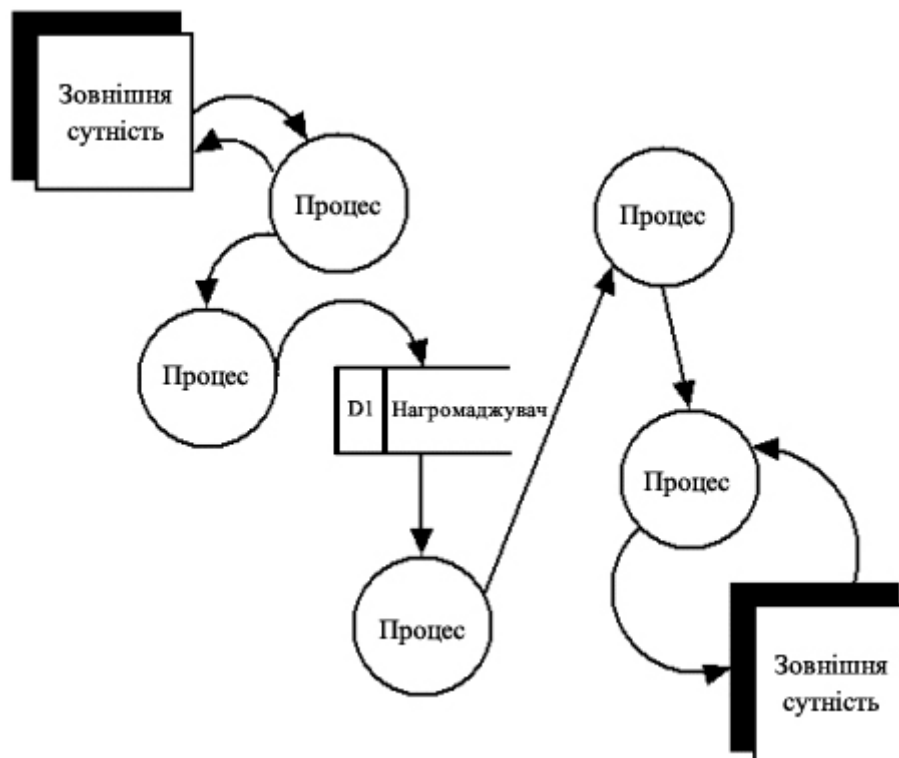


Рис. 3.18 – Приклад ДПД

Сховище (накопичувач) даних дозволяє на зазначених ділянках визначати дані, які будуть зберігатися в пам'яті між процесами. Фактично сховище представляє "зрізи" потоків даних у часі. Інформація, яку воно містить, може використовуватись в будь-який час після її одержання, при цьому дані можуть вибиратися в будь-якому порядку. Ім'я сховища повинно визначати його вміст і бути іменником.

1. Зовнішні сутності

Зовнішня сутність є матеріальним предметом або фізичною особою, яка виступає джерелом або приймачем інформації, наприклад, замовники, персонал, постачальники, клієнти, склад тощо. Визначення деякого об'єкту або системи як зовнішня сутність вказує на те, що вона знаходиться за межами аналізованої ІС. У процесі аналізу деякі зовнішні сутності можуть бути перенесені всередину діаграми аналізованої ІС, якщо це необхідно, або, навпаки, частина процесів ІС може бути винесена за межі діаграми і представлені як зовнішні сутності.

Зовнішня сутність позначається квадратом (рис. 3.3.2), який розташовується над діаграмою і відкидає на неї тінь, для того, щоб можна було виділити цей символ серед інших позначень:



Рис. 3.19 – Зовнішня сутність

2. Системи і підсистеми

При побудові моделі складної ІС вона може бути представлена у найзагальнішому вигляді на так званій контекстній діаграмі у вигляді однієї системи як єдиного цілого, або може бути декомпонована на низку підсистем.

Підсистема (або система) на контекстній діаграмі зображається наступним чином (рис. 3.20).



Рис. 3.20 – Підсистема

Номер підсистеми служить для її ідентифікації. У полі імені вводиться назва підсистеми у вигляді речення з підметом і відповідними визначеннями і доповненнями.

3. Процеси

Процес є перетворенням вхідних потоків даних у вихідні відповідно до певного алгоритму. Фізично процес може бути реалізований різними способами: це може бути підрозділ організації (відділ), що виконує обробку вхідних документів і випуск звітів, програма, апаратно реалізований логічний пристрій тощо.

Процес на діаграмі потоків даних зображається, як показано на рис. 3.21.



а)

б)

а) Нотація процесу за Gane and Sarson

б) Нотація процесу за Yourdon and Coad

Рис.3.21 – Процес

Номер процесу служить для його ідентифікації. У полі імені вводиться назва процесу у вигляді речення з активним недвозначним дієсловом у невизначеній формі (обчислити, розрахувати, перевірити, визначити, створити, отримати), за яким іде іменники в знахідному відмінку, наприклад:

- "Ввести відомості про клієнтів";

- "Видати інформацію про поточні витрати";
- "Перевірити кредитоспроможність клієнта".

4. Нагромаджувачі даних

Нагромаджувач даних може бути реалізований фізично у вигляді мікрофіші, шухляди у картотеці, таблиці в оперативній пам'яті, файлу на магнітному носії тощо. Нагромаджувач даних на діаграмі потоків даних зображається, як показано на рис. 3.22.

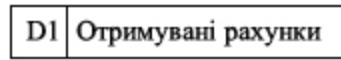


Рис. 3.22 – Нагромаджувач даних

Нагромаджувач даних ідентифікується буквою "D" і довільним числом. Назва нагромаджувача вибирається із міркування найбільшої інформативності для проектувальника.

5. Потоки даних

Потік даних визначає інформацію, яка передається через деяке з'єднання від джерела до приймача. Реальний потік даних може бути інформацією, яка передається по кабелю між двома пристроями, листами, що пересилаються поштою, магнітними стрічками або дискетами, які переносяться з одного комп'ютера на іншій тощо.

Потік даних на діаграмі зображається лінією, що закінчується стрілкою, яка показує напрямок потоку (рис. 3.23). Кожний потік даних має назву, що відображає його зміст.

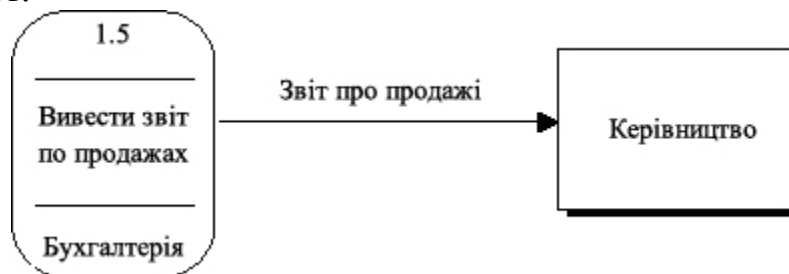


Рис. 3.23 – Потік даних

6. Побудова ієрархії діаграм потоків даних

Першим кроком при побудові ієрархії ДПД є побудова контекстних діаграм. Звичайно при проектуванні простих ІС будується єдина контекстна діаграма із зіркоподібною топологією, у центрі якої знаходиться так званий головний процес, який сполучений із приймачами і джерелами інформації, за допомогою яких з системою взаємодіють користувачі та інші зовнішні системи.

Якщо ж для складної системи обмежитися єдиною контекстною діаграмою, то вона міститиме дуже велику кількість джерел і приймачів інформації, які важко розташувати на листі паперу нормального формату, і крім того, єдиний головний процес не розкриває структури розподіленої системи. Ознаками складності (у значенні контексту) можуть бути:

- наявність великої кількості зовнішніх сутностей (десять і більше);
- розподілена природа системи;
- багатofункціональність системи із групою функцій, яка вже склалася або виявлюється, в окремі підсистеми.

Для складних ІС будується ієрархія контекстних діаграм. При цьому контекстна діаграма верхнього рівня містить не єдиний головний процес, а набір підсистем, сполучених потоками даних. Контекстні діаграми наступного рівня деталізують контекст і структуру підсистем.

Розробка контекстних діаграм вирішує проблему строгого визначення функціональної структури ІС на першій стадії її проектування, що особливо важливо для складних багатofункціональних систем, у розробці яких беруть

Для кожної підсистеми, присутньої на контекстних діаграмах, виконується її деталізація за допомогою ДПД. Кожний процес на ДПД, у свою чергу, може деталізуватися за допомогою ДПД або мініспецифікації. При деталізації повинні виконуватися наступні правила:

- правило балансування - означає, що при деталізації підсистеми або процесу деталізуюча діаграма у ролі зовнішніх джерел/приймачів даних може мати тільки ті компоненти (підсистеми, процеси, зовнішні сутності, нагромаджувачі даних), з якими має інформаційний зв'язок підсистема або процес на батьківській діаграмі;
- правило нумерації - означає, що при деталізації процесів повинна підтримуватися їхня ієрархічна нумерація. Наприклад, процеси, що деталізують процес з номером 12, одержують номери 12.1, 12.2, 12.3 і т.д.

Мініспецифікація є кінцевою вершиною ієрархії ДПД. Рішення про завершення деталізації процесу і використання мініспецифікації ухвалюється аналітиком виходячи з таких критеріїв:

- наявність у процесу відносно невеликої кількості вхідних і вихідних потоків даних (2-3 потоки);
- можливості опису перетворення даних процесом у вигляді послідовного алгоритму;
- виконання процесом єдиної логічної функції перетворення вхідної інформації у вихідну;
- можливості опису логіки процесу за допомогою мініспецифікації невеликого об'єму (не більше 20-30 рядків).

При побудові ієрархії ДПД переходити до деталізації процесів слід тільки після визначення змісту всіх потоків і нагромаджувачів даних, який описується за допомогою структур даних. Структури даних конструюються з елементів даних і можуть містити альтернативи, умовні входження і ітерації. Умовне входження означає, що даний компонент може бути відсутній у структурі. Альтернатива означає, що в структуру може входити один з перерахованих елементів. Ітерація означає входження будь-якої кількості елементів у вказаному діапазоні. Для кожного елемента даних може вказуватися його тип (безперервні або дискретні дані). Для безперервних даних може вказуватися одиниця вимірю-

вання (кг, см тощо), діапазон значень, точність уявлення і форма фізичного кодування. Для дискретних даних може вказуватися таблиця допустимих значень.

Після побудови закінченої моделі системи її необхідно верифікувати (перевірити на повноту і узгодженість). У повній моделі всі її об'єкти (підсистеми, процеси, потоки даних) повинні бути детально описані. Виявлені об'єкти, які не деталізуються, слід деталізувати, повернувшись на попередні кроки розробки. В узгодженій моделі для всіх потоків даних і нагромаджувачів даних повинно виконуватися правило збереження інформації: всі дані які поступають куди-небудь повинні бути прочитані, і всі прочитані дані повинні бути записані.

- накопичувачі даних (сховища).

Зовнішня сутність являє собою матеріальний об'єкт поза контекстом системи, що є джерелом або приймачем системних даних. Її ім'я повинно містити іменник, наприклад, "склад товарів". Передбачається, що об'єкти, представлені як зовнішні сутності, не повинні брати участь ні в якій обробці.

Процес побудови DFD

Процес побудови DFD починається зі створення так званої основної діаграми типу "зірка", на якій представлений моделюємий процес і всі зовнішні сутності, з якими він взаємодіє:

Рисунок «Діаграма типу «зірка»»

У випадку складного основного процесу він відразу представляється у вигляді декомпозиції на ряд взаємодіючих процесів.

Критерії складності методики потоків даних

- наявність великої кількості зовнішніх сутностей,
- багато-функціональність системи,
- її розподілений характер.

Повнота діаграми забезпечується, якщо в системі немає «висячих» процесів, не використовуваних у процесі перетворення вхідних потоків у вихідні.

Несуперечність системи забезпечується виконанням наборів формальних правил про можливі типи процесів:

на діаграмі не може бути потоку, що зв'язує дві зовнішні сутності - ця взаємодія видаляється з розгляду;

жодна сутність не може безпосередньо одержувати або віддавати інформацію в сховище даних - сховище даних є пасивним елементом, керованим за допомогою інтерфейсного процесу;

два сховища даних не можуть безпосередньо обмінюватися інформацією - ці сховища повинні бути об'єднані.

Переваги та недоліки

До переваг методики DFD відносяться:

- можливість однозначно визначити зовнішні сутності, аналізуючи потоки інформації усередині та поза системою;
- можливість проектування зверху вниз, що полегшує побудову моделі "як повинно бути";

- наявність специфікацій процесів нижнього рівня, що дозволяє здолати логічну незавершеність функціональної моделі та побудувати повну функціональну специфікацію розроблюваної системи.

До недоліків методики DFD віднесемо:

- необхідність штучного введення керуючих процесів, оскільки керуючі впливи (потоки) і керуючі процеси із точки зору DFD нічим не відрізняються від звичайних;

- відсутність поняття часу, тобто відсутність аналізу тимчасових проміжків при перетворенні даних (всі обмеження за часом повинні бути уведені в специфікаціях процесів).

3.4 ARIS

Для моделювання бізнес-процесів використовується декілька різних методів, в основі яких лежить як структурний, так і об'єктно-орієнтований підходи до моделювання. Проте, класифікація самих методів на структурні та об'єктні є доволі умовною, оскільки найбільш розвинуті методи використовують елементи обох підходів. Стисло розглянемо характеристики найбільш поширених методів:

- метод функціонального моделювання SADT (IDEF0);
- метод моделювання процесів IDEF3;
- моделювання потоків даних DFD;
- метод ARIS;
- метод Ericsson-Penker;
- метод технології Rational Unified Process.

Метод SADT (Structured Analysis and Design Technique) вважається класичним методом підходу до управління на основі процесів, базовим принципом якого є структуризація діяльності організації у відповідності з її бізнес-процесами. Бізнес-модель відповідає таким вимогам:

- верхній рівень моделі відображає виключно контекст системи - взаємодію підприємства із зовнішнім середовищем;
- другий рівень описує основні види діяльності підприємства - тематично згруповані бізнес-процеси;
- подальша деталізація бізнес-процесів здійснюється за допомогою бізнес-функцій та елементарних бізнес-операцій, згрупованих за певними ознаками;
- опис елементарної бізнес-операції відбувається шляхом визначення алгоритму її виконання.

Метод моделювання IDEF3 - частина сімейства стандартів IDEF; використовується для моделювання послідовності виконання дій і їх взаємозалежностей в рамках процесу. Метод отримав визнання серед системних аналітиків як доповнення до методу функціонального моделювання IDEF0.

Основою моделі IDEF3 служить сценарій процесу, який відокремлює послідовність дій і підпроцесів системи. Як і в методі IDEF0, основною одиницею

моделі є діаграма. Іншим важливим компонентом є дія або "одиниця роботи" (Unit of Work), взаємодія яких зображається за допомогою зв'язків.

Діаграми потоків даних (Data Flow Diagrams - DFD) представляють собою ієрархію функціональних процесів, що пов'язані потоками даних. Мета такого представлення полягає у демонстрації того, як кожен процес перетворює свої вхідні дані у вихідні і виявлення зв'язків між цими процесами.

Відповідно до методу, модель системи визначається як ієрархія діаграм потоків даних, основними компонентами яких є:

- зовнішні об'єкти;
- системи та підсистеми;
- процеси;
- накопичувачі даних;
- потоки даних.

Метод ARIS (Architecture of Integrated Information System), представляє собою комплекс засобів аналізу і моделювання діяльності підприємства. Його методичну основу складає сукупність різноманітних методів моделювання, що відображають різні погляди на системи. ARIS підтримує чотири типи моделей, які віддзеркалюють різні аспекти системи, що досліджується:

- організаційні, що представляють структуру системи;
- функціональні, які містять ієрархію цілей;
- інформаційні - відображають структуру всієї інформації, необхідної для реалізації функцій системи;
- моделі управління, що представляють комплексний підхід до реалізації бізнес-процесів в рамках системи.

Для побудови зазначених типів моделей використовуються як власні методи моделювання ARIS, так і різні відомі методи та мови моделювання, зокрема UML.

Автори методу Ericsson-Penker створили свій профіль UML для моделювання бізнес-процесів - Ericsson-Penker Business Extensions, ввівши набір стереотипів, які описують основні категорії бізнес-моделі: процеси, ресурси, правила і цілі діяльності підприємства.

Мова UML використовується також в методі, який є частиною технології Rational Unified Process (фірми IBM). Цей метод спрямовано насамперед на створення основи для формування вимог до ПЗ. Передбачає побудову двох базових моделей:

- моделі бізнес-процесів (Business Use Case Model);
- моделі бізнес-аналізу (Business Analysis Model). Business Use Case визначається як опис послідовності дій (поток) в рамках певного бізнес-процесу, що дає результат для певної діючої особи.

3.5 Метод технології Rational Unified Process (UML)

Rational Unified Process (RUP) є ітеративним процесом розробки програмного забезпечення створеним Rational Software — підрозділом IBM з 2003. RUP не є єдиним, конкретним розпорядчим процесом, а скоріше фреймворком процесу, що має бути адаптованим організаціями які займаються розробкою та командами розробників які оберуть елементи процесу, які підходять під їх потреби.

о 1997 року, Rational придбав Verdex, Objectory, Requisite, SQA, Performance Awareness, та Pure-Atria. Поєднання баз досвіду цих компаній привело до вироблення семи «найкращих практик» сучасної програмної інженерії:

1. Розробляти ітеративно, керуючись ризиками.
2. Управляти вимогами
3. Використовувати компонентну архітектуру
4. Моделювати програмне забезпечення візуально
5. Постійно перевіряти якість
6. Контролювати зміни
7. Підлаштовуватись

RUP пропонує ітеративний підхід до проектування та розробки ПС, заснований на спіральному життєвому циклі. Весь життєвий цикл включає чотири фази - входження в проект (дослідження), розвиток (уточнення плану), конструювання та розгортання. Кожна фаза складається з послідовності ітерацій, число яких може бути будь-яким.

У кожній ітерації перераховані вище технологічні процеси послідовно застосовуються до розробки невеликої частини ПС. При цьому допустимо пред'явлення результату замовникові. Він має можливість оцінити виконану реалізацію, видати свої зауваження, які можуть привести до зміни та уточнення вимог до ПС. Наступна ітерація передбачає розширення вже розробленої частини шляхом реалізації та інтеграції чергової порції вимог і обліку зміни вимог відповідно до зауважень замовника. Така організація процесу має цілий ряд переваг.

- Зміни та уточнення вимог виявляються вже в ранній період розробки, коли обсяг програмного коду порівняно невеликий, тому трудомісткість внесення змін істотно нижче.

- Вже на ранній стадії процесу розробки є можливість залучення фахівців замовника для оцінки проміжних версій (прототипів) ПС. Як результат - значно більш висока ймовірність того, що кінцевий продукт буде робити саме те, чого чекає від нього замовник, тобто, гарантується висока якість ПС.

-Знижуються архітектурні та інтеграційні ризики. При ітеративном підході створюється стійка архітектура, яка опрацьовується на стадіях дослідження, а потім перевіряється і уточнюється в декількох початкових ітераціях. Кожна ітерація передбачає інтеграцію нових елементів в систему, тобто, кількість інтегрованих елементів в кожній ітерації невелика і легше простежується, ніж при глобальній інтеграції всіх розроблених елементів ПС.

- Ітеративний підхід сприяє більш повному повторному використанню програмних елементів. Аналіз результатів кожної ітерації дозволяє архітекторам ПС виділити фрагменти, потенційно підлягають повторному використанню, а в наступній ітерації оформити їх як повторно використовувані коди. Це властивість дуже добре поєднується з ідеями ГО і КБ проектування.

RUP - процес, що направляється варіантами використання

Поняття use case, введене в мові UML, є основою для виконання всіх етапів життєвого циклу, що розглядаються в RUP. Поняття «business use case» (вид діяльності) є ключовим при бізнес-аналізі. На етапах аналізу вимог, проектування і реалізації use cases виступають в якості варіантів використання системи (ВІ), будучи тією «грубкою», від якої «танцюють» аналітики і розробники при виконанні проектування і реалізації ПС. При аналізі вимог, виділивши ВІ, ми тим самим визначаємо вимога або рівень ієрархії вимог до ПС. При деталізації ВІ визначаються об'єкти, і способи їх взаємодії, які повинні бути реалізовані в програмному коді.

Слід особливо наголосити роль ВІ в плануванні ітерацій. Як визначити, яка функціональність повинна бути реалізована в черговий ітерації? Тут на виручку приходять діаграми ВІ. Кожному ВІ можна приписати пріоритет, який визначає, в якій ітерації його слід реалізувати. Можна показати все ВІ, що реалізуються в черговий ітерації, на окремій діаграмі (або декількох діаграмах).

У RUP визначено 9 технологічних процесів, для кожного з яких запропонована методика виконання. Технологічні процеси діляться на дві категорії - основні процеси та процеси підтримки. До основних відносяться:

- Бізнес-аналіз,
- Управління вимогами,
- Аналіз і проектування,
- Реалізація,
- Тестування,
- Розгортання.

Допоміжні процеси включають:

- Управління проектом,
- Управління конфігурацією,
- Управління середовищем

Для кожного технологічного процесу передбачені ролі, що визначають поведінку і обов'язки окремих осіб і груп, які працюють в одній команді (наприклад, системний аналітик, тестувальник), види діяльності, що визначають роботи, що виконуються виконавцями (наприклад, проектування класу, проектування ВІ) і артефакти - документи, використовувані, породжувані або модифікуються процесом. Основні артефакти в RUP - модель, елемент моделі, документ, вихідний код, що виконується програма.

3.6 Метод Ericsson-Penke

Механізми призначені UML для того, щоб розробники могли адаптувати мову моделювання до своїх конкретних потреб, не змінюючи при цьому його модель. Мова UML принципово відрізняється від таких засобів моделювання, як IDEF0, IDEF1X, IDEF3, DFD та ін. Перераховані мови моделювання можна визначити як сильно типізовані (по аналогії з мовами програмування), оскільки вони не допускають довільній інтерпретації елементів моделей. UML, допускаючи таку інтерпретацію є слабо універсальна мова.

Метод Ericsson-Penker використовується при проектуванні проектів із застосуванням UML в рамках процесного підходу до моделювання бізнес-процесів. Автори методу створили свій профіль UML для моделювання бізнес-процесів, ввівши набір стереотипів, що описують процеси, ресурси, правила і цілі діяльності організації. Метод використовує чотири основні категорії бізнес-моделі:

Ресурси - різні об'єкти, що використовуються або беруть участь в бізнес-процесах (люди, матеріали, інформація або продукти).

Процеси - види діяльності, які змінюють стан ресурсів відповідно до бізнес-правилами.

Цілі - призначення бізнес-процесів мети можуть бути розбиті на підцілі і співвіднесені з окремими процесами.

Бізнес-правила - умови або обмеження виконання процесів (функціональні, поведінкові або структурні). Правила можуть бути визначені з використанням мови OCL.

Основною діаграмою UML, використовуваної в даному методі, є діаграма діяльності. Метод Eriksson-Penker представляє процес у вигляді діяльності зі стереотипом "process" (основою даного подання є розширення методу IDEF0). Повна бізнес-модель включає безліч уявлень, подібних уявленням архітектури ПЗ. Кожна вистава виражено в одній або більше діаграмах UML. Діаграми можуть мати різні типи і зображати процеси, правила, цілі і ресурси у взаємодіях один з одним. Метод використовує чотири різних уявлення бізнес-моделі:

- концептуальне уявлення - структура цілей і проблем;
- уявлення процесів - взаємодія між процесами і ресурсами (у вигляді набору діаграм діяльності);
- структурне уявлення - структура організації і ресурсів (у вигляді діаграм класів);
- уявлення поведінки - поведінка окремих ресурсів і деталізація процесів (у вигляді діаграм діяльності, станів і взаємодії).

3.7 Основні принципи та розвиток моделювання бізнес-процесів

Моделювання бізнес-процесів в компанії може бути направлено на рішення великого числа різних завдань:

- Точно визначити результат бізнес-процесу і оцінити його значення для бізнесу.

- Визначити набір дій, що становлять бізнес-процес. Чітке визначення набору завдань і дій, які необхідно виконати, надзвичайно важливо для детального розуміння процесу.

- Визначити порядок виконання дій. Дії в рамках одного бізнес-процесу можуть виконуватися як послідовно, так і паралельно. Очевидно, що паралельне виконання, якщо воно допустимо, дозволяє скоротити загальний час виконання процесу і, отже, підвищити його ефективність.

- Провести поділ зон відповідальності: визначити, а потім відстежувати, який співробітник або підрозділ компанії несе відповідальність за виконання того чи іншої дії або процесу в цілому.

- Визначити ресурси, що споживаються бізнес-процесом. Точно знаючи, хто які ресурси використовує і для яких операцій, можна підвищити ефективність використання ресурсів за допомогою планування і оптимізації.

- Зрозуміти суть взаємодій між що беруть участь в процесі співробітниками і підрозділами компанії і оцінити, а потім підвищити ефективність комунікації між ними.

- Побачити рух документів в ході процесу. Бізнес-процеси виробляють і споживають різні документи (в паперовій або електронній формі). Важливо розібратися, звідки і куди йдуть документи або інформаційні потоки, і визначити, оптимально їх рух і чи дійсно всі вони необхідні.

- Визначити потенційні вузькі місця і можливості для поліпшення процесу, які будуть використані пізніше для його оптимізації.

- Більш ефективно впровадити стандарти якості, наприклад ISO 9000, і успішно пройти сертифікацію.

- Використовувати моделі бізнес-процесів в якості керівництва для нових співробітників.

- Ефективно провести автоматизацію бізнес-процесів в цілому або окремих їхніх кроків, включаючи автоматизацію взаємодії із зовнішнім середовищем - клієнтами, постачальниками, партнерами.

- Розібравшись в сукупності бізнес-процесів компанії, зрозуміти і описати діяльність підприємства в цілому.

Після визначення результату слід розібратися з послідовністю дій, складових процес. Послідовність дій моделюється на різних рівнях абстракції. На самому верхньому рівні показують тільки найбільш важливі кроки процесу (Зазвичай не більше десяти). Потім проводиться декомпозиція кожного з високорівневих кроків (підпроцесів

На основі зібраної інформації будується модель звичайного, або оптимального, виконання процесу і визначаються можливі сценарії його виконання зі збоями. різні збої(Виняткові ситуації - виключення) можуть порушувати оптимальний хід процесу, тому слід вказати, яким чином виключення будуть «оброблені», тобто які дії робляться в разі виникнення виняткової ситуації.

На рис. 3.24 показані основні кроки при побудові моделі бізнес-процесу.



Рис. 1.2. Шаги построения модели бизнес-процесса

Рис. 3.24 - Основні кроки при побудові моделі бізнес-процесу

Важливою частиною побудови моделі бізнес-процесу є дослідження аспектів його ефективності. Сюди входять використання ресурсів, час виконання робіт співробітниками, можливі затримки і простой. Необхідно розробити систему показників, або метрик, для оцінки ефективності процесу. Частково в якості метрик можуть бути взяті використовувані в компанії КРІ (Key Performance Indicator), однак можуть знадобитися і додаткові характеризують даний процес показники.

Необхідно виявити події, які можуть перервати хід процесу. В разі переривання може знадобитися коректно «відкотити» (компенсувати) ті кроки процесу, які вже були виконані. Для цього слід визначити логіку компенсуючих дій для кожного перериває події.

Нарешті, слід розглянути наявні програмні засоби, що здійснюють підтримку бізнес-процесу. Це важливо, тому що програмне забезпечення може приховувати деякі особливості поведінки процесу, не в повній мірі відомі виконуючим окремі кроки співробітникам. Зібрана на цьому етапі інформація буде корисна при подальшій автоматизації процесу.

Зібравши всі зазначені відомості, можна отримати гарне уявлення про хід бізнес процесу. На етапі моделювання повинні бути отримані наступні результати:

- Процесна карта, що показує зв'язок між різними бізнес-процесами і їх взаємодії. На процесній карті, як правило, кожен бізнес-процес компанії зображений у вигляді прямокутника, стрілками показані зв'язку між ними (наприклад, залежність одного процесу від іншого, або заміна одного процесу іншим при виконанні деякого умови), а також представлені різні документи, які передаються з процесу в процес або регламентують їх хід (стандарти, інструкції тощо).

- Діаграма ролей, що показує ролі при виконанні процесу і зв'язку між ними. Діаграма ролей не є ієрархічною. Вона являє такі зв'язки, як участь в групі, керівництво, комунікацію, заміщення однієї ролі іншого і т. д.

- Модель «як є» кожного розглянутого бізнес-процесу, детально описує процес і відображає хід процесу, дії, ролі, рух документів, а також точки можливої оптимізації. Така модель включає в себе:

- діаграму оточення процесу, що представляє бізнес-процес у вигляді одного дії (тобто не розкриває хід процесу), для якого можуть бути показані запускання процесу подія, необхідні вхідні дані, результат, ролі, показники ефективності, переривають події і компенсуючі процеси, регламентуючі документи, пов'язані бізнес-цілі;

- високорівневу діаграму процесу, що показує його великі кроки (Зазвичай не більше десяти) і пов'язані з ними ролі;

- докладні діаграми для кожного кроку високорівневою моделі (в залежно від складності процесу тут може використовуватися кілька ієрархічно організованих діаграм), в деталях показують хід процесу, переривають події, бізнес-правила, ролі і документи;

- діаграму обробки винятків, що показує, які дії виконуються в разі даної виняткової ситуації і ким, а також куди передається керування після закінчення обробки виключення.

Контрольні питання?

- 1) Які переваги функціонального моделювання бізнес-процесів (IDEF0)?
- 2) Опишіть загальні принципи методології IDEF0?
- 3) Яким чином можна графічно представити структури в IDEF0?
- 4) Назвіть приклади використання IDEF0?
- 5) Які переваги моделювання бізнес-процесів IDEF3)?
- 6) Опишіть загальні принципи методології IDEF3?
- 7) Назвіть приклади використання IDEF0?
- 8) Наведіть приклади використання формалізованого опису бізнес-процесів?
- 9) В чому полягає суть моделювання потоків даних (DFD)?
- 10) Особливості використання методу ARIS?
- 11) Які переваги та недоліки має метод технології Rational Unified Process (UML)?
- 12) Які переваги методу Ericsson-Penke?
- 13) Назвіть основні принципи моделювання бізнес-процесів?
- 14) Які перспективи розвитку моделювання бізнес-процесів?

Глосарій

Аналіз (від грец. *αναλυσις* — «розклад») — розчленування предмету пізнання, абстрагування його окремих сторін чи аспектів. Метод дослідження, який вивчає предмет, уявно чи реально розчленовуючи його на складові елементи, як-от частини об'єкта, його ознаки, властивості, відношення, відтак розглядає кожен з виділених елементів окремо в межах єдиного цілого; протилежний метод — синтез.

Моделювання — це метод дослідження різних явищ і процесів, вироблення варіантів управлінських рішень. Моделювання ґрунтується на заміщенні реальних об'єктів їх умовними зразками, аналогами. Методом моделювання описуються структура об'єкта (статична модель), процес його функціонування і розвитку (динамічна модель).

Бізнес-процес - будь-яка діяльність, що має вхідний продукт, додає вартість до нього, та забезпечує вихідний продукт для внутрішнього або зовнішнього споживача. Існують три види бізнес-процесів:

процеси управління — бізнес-процеси, які управляють функціонуванням системи. Прикладом керувального процесу може служити корпоративне управління та стратегічний менеджмент.

основні — бізнес-процеси, які складають основний бізнес компанії і створюють основний потік доходів. Прикладами операційних бізнес-процесів є постачання, виробництво, маркетинг та збут.

забезпечувальні — бізнес-процеси, які обслуговують основний бізнес. Наприклад, бухгалтерський облік, кадрове, інформаційне забезпечення.

Бізнес-процес починається з попиту споживача і закінчується його задоволенням.

Геоінформаційна система (ГІС) (GIS) — це сукупність електронних карт з умовними позначеннями об'єктів на них, баз даних з інформацією про ці об'єкти та програмного забезпечення для зручної роботи з картами і базами як з єдиним цілим.

Бази даних (Databases) — це структурований набір даних про певні характеристики фізичних чи віртуальних систем.

Управління — Перетворення інформації про стан об'єкта в командну інформацію від суб'єкта. Це цілеспрямований програмований чи довільний вплив на об'єкти задля досягнення кінцевої мети за допомогою процесорів, явищ, процесів, коли є з ними взаємодія в режимі детермінованої чи довільної програми/регламенту. Управління проектом/об'єктом-системою, її компонентами та процесами, з метою підвищення ефективності функціонування систем відбувається ще на етапі системного проектування, створення/утворення, формування, розвитку, становлення, функціонування/життя системи. Ефективність управління визначається адекватністю дій управління що до об'єкта управління. Управління є першим етапом тактичного рівня в алгоритмі системно-організаційній діяльності. Управління є координуючим, адміністративним, ви-

конавчим рівнем в тактиці. Стратегічне управління пов'язане з аналізом проблем, виявленням крайових умов задач, пошуку оптимальних моделей рішень, забезпечення виконавчими процесорами, ресурсами, програмою дій, в тому числі керівними процесорами чи кадровими управлінцями для тактичного, виконавчого рівня проекту/об'єкта-системи.

ІТ-проект - це процес створення продукту. Це те, що робить команда, щоб видати замовнику продукт. Продукт - те, що хоче отримати замовник. Продукт – результат (чи набір результатів) поставки по контракту. Метою будь-якого проекту є отримання кінцевого продукту.

Інформаційна система - сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

Інформаційна технологія - сукупність методів, виробничих процесів і програмно-технічних засобів, інтегрованих з метою збирання, опрацювання, зберігання, розповсюдження, показу і використання інформації в інтересах її користувачів. Технології, що забезпечують та підтримують інформаційні процеси, тобто процеси пошуку, збору, передачі, збереження, накопичення, тиражування інформації та процедури доступу до неї.

Геоінформаційні технології (ГІС-технології) (GIS technology) — технологічна основа створення географічних інформаційних систем, які дозволяють реалізувати функціональні можливості ГІС.

ГІС-аналіз (GIS analysis) — це процес пошуку географічних закономірностей в даних і взаємовідношень між просторовими об'єктами

База геоданих (Geodatabases) представляє собою відкриту структуру для збереження і керування даними ГІС (просторова геометрія, таблиці, зображення), реалізовану в системі керування реляційними базами даних.

Система (System) – це сукупність елементів, певним чином пов'язаних і взаємодіючих між собою для виконання заданих цільових функцій.

Цифрова карта (Digital map) – цифрова модель місцевості, створена шляхом цифрування картографічних джерел, фотограмметричної обробки даних дистанційного зондування, цифрової реєстрації даних польових зйомок або іншим способом.

Електронна карта (Electronic map) – картографічне зображення, яке візуалізоване на дисплеї (відеоекрані) комп'ютера на основі даних цифрових карт або баз даних ГІС з використанням програмних і технічних засобів у прийнятій для карт проекції і системі умовних знаків.

Геометрична мережа (Geometric network) є набором просторових об'єктів, що моделюють ребра мережі і з'єднання мережі.

Література

1. Маркина Т.А. Управление проектами в информационных технологиях. Учебное пособие. – СПб: Университет ИТМО, 2016. – 88 с.
2. Деренская Я.Н. Классификация проектов в проектном менеджменте. [Электронный ресурс]. – Режим доступа: <http://www.finanalys.ru/litra/328/2895.html>
3. Дружинін Є.А. Методологічні основи ризик-орієнтованого підходу до управління ресурсами проектів і програм розвитку техніки: Автореф. дис... д-ра техн. наук: 05.13.22 / Є.А. Дружинін ; Нац.аерокосм. ун-т ім. М.С.Жуковського "Харк. авіац. ін-т". — Х., 2006. — 34 с.
4. Заговора О.В., Концевич В.Г. Учёт особенностей ИТ-проектов при определении их жизненного цикла. [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/uchet-osobennostey-it-proektov-pri-opredelenii-ih-zhiznennogo-tsikla>
5. Катренко А.В. Методи управління ризиками в ІТ-проектах / А. В. Катренко, І. В. Рішняк // Комп'ютерні науки та інформаційні технології (CSIT-2008): III Міжнар. наук.- практ. конф., 25–27 вересня 2008р.: тези доповіді – Львів, 2008. – С. 245–247.
6. Коньшунова А.Ю. К вопросу о классификации проектов в проектном управлении. [Электронный ресурс]. – Режим доступа: <http://sibac.info/11760>
7. Листер Т. Вальсируя с медведями. Управление рисками в проектах по разработке программного обеспечения./ Листер Т., ДеМарко Т. – М: Компания р.т. Office, 2005. – 322 с.
8. Модель управління проектними ризиками / І.В. Рішняк // Комп'ютерні системи проектування. Теорія і практика. Вісник Національного університету "Львівська політехніка". – 2004. – № 522. – С.155–160.
9. Молоткова Н.В., Сахаров И.С. Инфраструктура управления качеством ИТ-проектов. [Электронный ресурс]. – Режим доступа: http://vernadsky.tstu.ru/pdf/2008/03g/22g_13.pdf
10. Песоцкая Е.Ю. Управление рисками при внедрении ИТ-проектов // Компьютерное моделирование в науке и технике. [электронный ресурс] – режим доступа: <http://econfr.ae.ru/article/3910>.
11. Руководство к своду знаний по управлению проектами (Руководство РМВоК) – 5-е изд., США: Project Management Institute, 2013. – 587 с.
12. Управління ризиками в проектній діяльності / О.М. Верес, А.В. Катренко, І.В. Рішняк, В.М. Чаплига // Інформаційні системи та мережі // Вісн. Нац. ун-ту "Львівська політехніка". – 2003. – № 489. – С. 38–49.
13. Плескач В. Л. Інформаційні системи і технології на підприємствах : підручник / В. Л. Плескач, Т. Г. Затонацька. – К. : Знання, 2011. – 718 с.
14. Project management institute. [Електронний ресурс] // Режим доступу: <http://www.pmi.org/>.

15. Стандарти управління проектами. [Електронний ресурс] // Режим доступу:
http://studme.org/1055120821033/menedzhment/standarty_upravleniya_proektami.
16. Каскадна модель. [Електронний ресурс] // Режим доступу:
<http://asset.in.ua/novosti-ukrainy/item/13713-1453915397>.
17. RAD програмування. [Електронний ресурс] // Режим доступу:
[https://ru.wikipedia.org/wiki/RAD_\(програмування\)](https://ru.wikipedia.org/wiki/RAD_(програмування)).
18. Документація проекту. [Електронний ресурс] // Режим доступу:
<http://www.pmphelp.net/index.php?id=33>.

Навчальне видання

**Крижановський Євгеній Миколайович
Ящолт Андрій Русланович
Козачко Олексій Миколайович
Жуков Сергій Олександрович**

**МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ ТА УПРАВЛІННЯ
ІТ-ПРОЕКТАМИ**

Електронний навчальний посібник

Оригінал-макет підготовлено А. Р. Ящолтом