

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Ушакова І. О.

**ОСНОВИ СИСТЕМНОГО АНАЛІЗУ
ОБ'ЄКТІВ І ПРОЦЕСІВ КОМП'ЮТЕРИЗАЦІЇ**

Навчальний посібник

Частина 2

Харків. Вид. ХНЕУ, 2008

УДК004.4(042.4)
ББК32.973 – 018я7
А00

Рецензенти: докт. техн. наук, професор кафедри ІУС Харківського національного університету радіоелектроніки *Авраменко В. П.*; канд. техн. наук, доцент, зав. кафедри інформаційної і комп'ютерної техніки Харківського національного економічного університету *Степанов В. П.*

Рекомендовано до видання рішенням вченої ради Харківського національного економічного університету.

Протокол №4 від 27.11.2007 р.

Авторський колектив:

Ушакова І. О.

А00 Основи системного аналізу об'єктів і процесів комп'ютеризації.
Ч. 2. Навчальний посібник для студентів напрямку "Комп'ютерні науки"
/ І. О. Ушакова. — Харків: Вид. ХНЕУ, 2008. — 308 с. (Укр. мов.)

Розкрито теоретичні основи проектування ІС з точки зору моделей їх життєвого циклу. Особлива увага приділяється основам програмної інженерії. Розглянуті чинні стандарти в цій галузі, а також особливості різних моделей життєвого циклу програмного забезпечення: від класичної каскадної моделі до моделей, які підтримують Agile-технології. Докладно висвітлюється технологія канонічного проектування. Проаналізовані сучасні підходи до проектування, значна увага приділяється комбінованому процесно-орієнтованому підходу, інструментальним засобам, що підтримують цей підхід, і, зокрема, інтегрованій системі ARIS. Розглянуті питання стосовно типового проектування ІС. Висвітлюються особливості організації та управління проектами.

Розраховано на студентів напрямку "Комп'ютерні науки" спеціальностей "Інформаційні управляючі системи і технології", "Комп'ютерний еколого-економічний моніторинг".

ISBN 0-0-0-0

**УДК 004.4(042.4)
ББК 32.973 – 018я7**

© Харківський національний економічний університет, 2008

© Ушакова І. О.
2008

Вступ

Створення ефективних систем управління організаціями – одна з першорядних проблем сьогодення. Питання, що пов'язані зі збиранням, зберіганням, пошуком, обробкою, перетворенням, розповсюдженням і використанням інформації в організації, мають вирішальне значення для ефективної її діяльності. Тому інформаційні системи управління організаціями розвиваються дуже стрімко і потребують для їх створення фахівців відповідної кваліфікації.

Ускладнення сучасних інформаційних систем (ІС) вимагає використання ефективних технологій їх проектування, які прискорюють створення, впровадження і розвиток проектів ІС, підвищують їх надійність, сприяють їх адаптуванню до змін в навколишньому середовищу. У навчальному посібнику саме висвітлені питання теорії та практики створення інформаційних систем управління організаціями, а також організації та управління процесами проектування ІС за допомогою різноманітних методів і сучасних інструментальних засобів.

Друга частина навчального посібника є основою викладання курсу "Основи системного аналізу об'єктів і процесів комп'ютеризації" для студентів спеціальностей "Інформаційні управляючі системи і технології", "Комп'ютерний еколого-економічний моніторинг" усіх форм навчання. Він охоплює 3-й і 4-й модулі курсу.

У третьому модулі розглянуті теоретичні основи проектування ІС з точки зору моделей життєвого циклу ІС. Наведені визначення основних понять технології проектування, розглянутий її формалізований опис, методи та засоби проектування. Особлива увага приділяється основам програмної інженерії. Розглянуті чинні стандарти в цій галузі, а також особливості різних моделей життєвого циклу програмного забезпечення: від класичної каскадної моделі до моделей, які підтримують Agile-технології. Докладно висвітлюється технологія канонічного проектування за ГОСТ серії 34.

У четвертому модулі розглянуті індустріальні методи проектування ІС. Проаналізовані сучасні підходи до проектування, значна увага приділяється комбінованому процесно-орієнтованому підходу, інструментальним засобам, що підтримують цей підхід, і, зокрема, інтегрованій системі ARIS. Розглянуті питання стосовно типового проектування ІС, особливо параметричного і модельного об'єктного типового проектування. Висвітлюються особливості організації та управління проектами.

Модуль 3. Теоретичні основи проектування ІС

8. Методологія проектування ІС

Методологія проектування складає основу проекту будь-якої ІС. Методологія проектування реалізується за допомогою конкретних технологій, стандартів, що їх підтримують, методів та інструментальних за-собів.

8.1. Технологія проектування ІС

Сучасні інформаційні технології пропонують широкий набір способів реалізації ІС, вибір яких здійснюється на основі вимог з боку майбутніх користувачів. Потреба у створенні ІС може обумовлюватися необхідністю автоматизації чи модернізації існуючих інформаційних процесів, або необхідністю корінної реорганізації в діяльності організації (проведенні бізнес-реінжинірингу). Потреба у створенні ІС вказує на те, по-перше, для досягнення якої саме мети необхідно розробити систему; по-друге, до якого моменту часу доцільно здійснити розробку; по-третє, які витрати необхідно здійснити для проектування системи.

Мета проектування ІС – створити проект інформаційної системи, який становить технічна документація з докладним описом усіх проектних рішень щодо створення й експлуатації ІС.

Для теорії ухвалення рішень **процес проектування ІС** – це процес ухвалення проектно-конструкторських рішень, спрямованих на одержання опису системи (проекту ІС), що задовольняє вимоги замовника.

Під **проектом ІС** слід розуміти проектно-конструкторську і технологічну документацію, в якій наведений опис проектних рішень щодо створення й експлуатації ІС у конкретному програмно-технічному середовищі.

Під **проектуванням ІС** мається на увазі процес перетворення вхідної інформації про об'єкт проектування, про методи проектування і про досвід проектування об'єктів аналогічного призначення в проект ІС відповідно до державних стандартів (рис. 1). З цього погляду проектування ІС зводиться до послідовної формалізації проектних

рішень на різних стадіях життєвого циклу ІС: планування й аналізу вимог, технічного та робочого проектування, впровадження й експлуатації ІС.

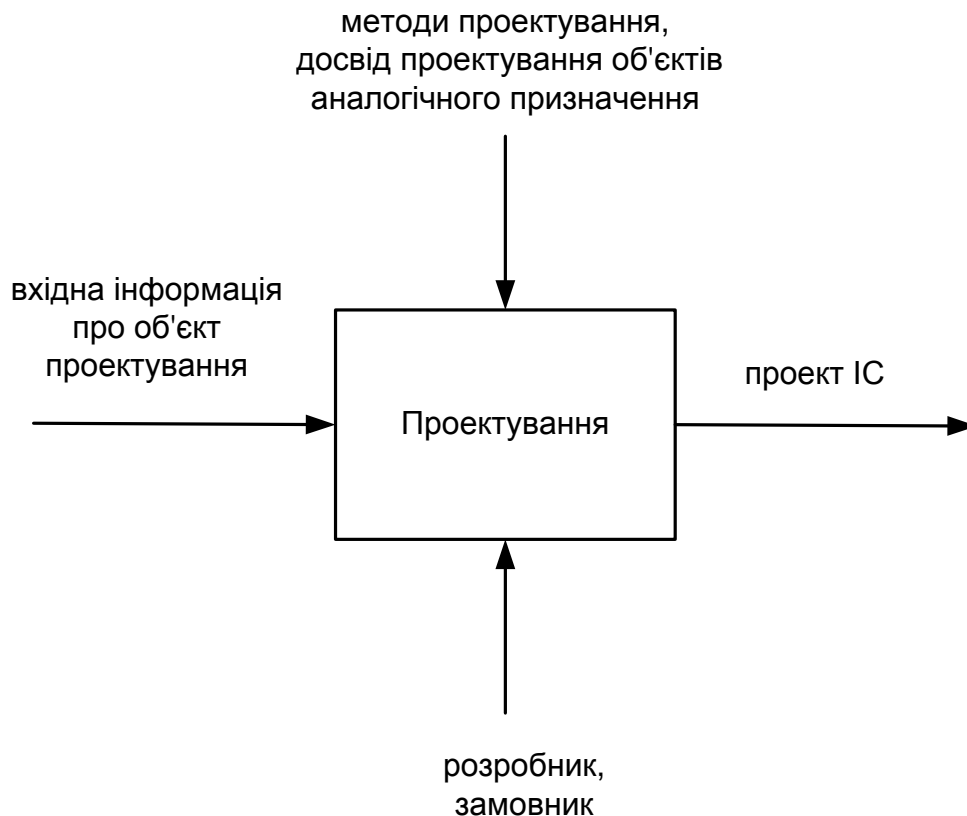


Рис. 1. Проектування ІС

Об'єктами проектування ІС є окремі елементи або комплекси їх функціональних і забезпечувальних частин. Так, функціональними елементами відповідно до традиційної декомпозиції є завдання, комплекси завдань і функції управління. У складі забезпечувальних частин ІС об'єктами проектування слугують елементи і комплекси інформаційного, програмного та технічного забезпечення системи.

Предметною областю проектування ІС є система організаційно-економічного управління діяльністю підприємства.

Суб'єктами проектування ІС є колективи фахівців, що здійснюють проектну діяльність, а саме:

організація-розробник, як правило, це спеціалізована (проектна) організація;

організація-замовник, для якої необхідно розробити ІС.

Масштаби розроблюваних систем визначають склад і кількість учасників процесу проектування. За великого обсягу й жорстких строків виконання проектних робіт у розробці системи може брати участь декілька проектних

колективів (організацій-розробників). У цьому випадку виділяється головна організація, що координує діяльність всіх організацій-співвиконавців.

Форма участі співвиконавців у розробці проекту системи може бути різною. Найбільш розповсюдженою є форма, за якої кожен співвиконавець виконує проектні роботи від початку до кінця для якої-небудь частини розроблюваної системи. Зазвичай це буває функціональна підсистема або взаємозалежний комплекс завдань управління. Рідше зустрічається форма участі співвиконавців, за якої окремі співвиконавці виконують роботи на окремих етапах процесу проектування. Можливий варіант, за якого функції замовника і розробника поєднуються. У такому випадку ІС проектується власними силами.

Організація проектування передбачає визначення методів взаємодії проектувальників між собою і з замовником у процесі створення проекту ІС, що можуть також підтримуватися набором специфічних засобів.

Проектування ІС передбачає використання проектувальниками певної технології проектування, що відповідає масштабу й особливостям розроблюваного проекту.

Технологія проектування ІС – це сукупність методів і засобів проектування ІС, організації проектування (рис. 2).

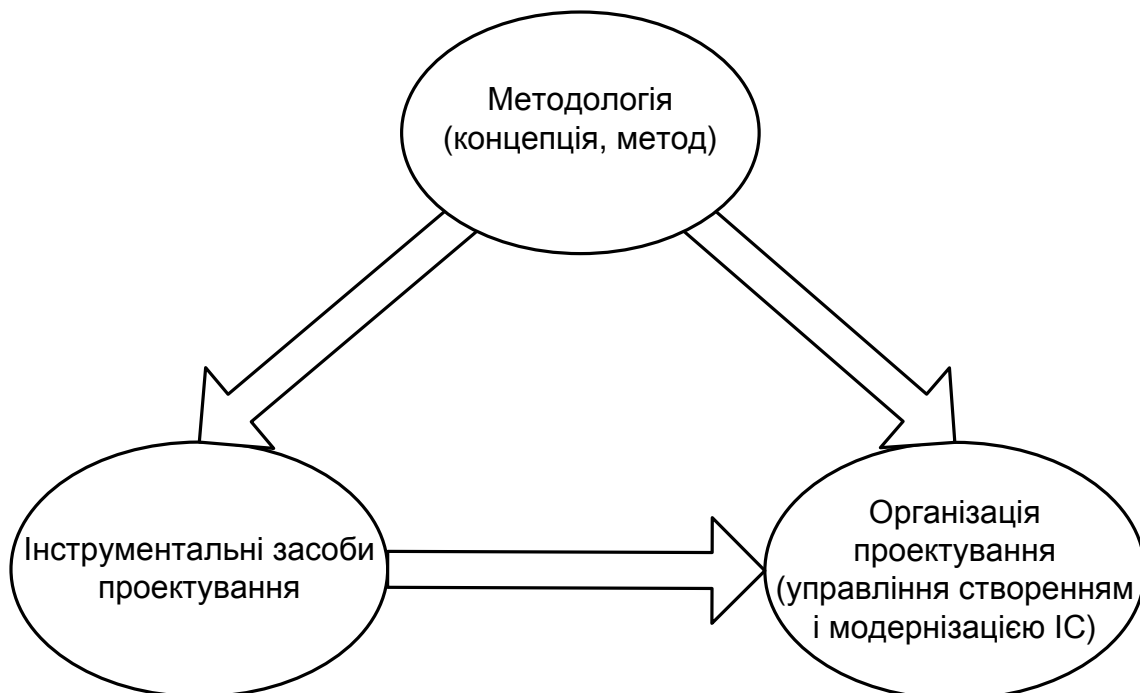


Рис. 2. Компоненти технології проектування

Технологію проектування можна розглядати як сукупність наступних складових:

покрокової процедури, яка визначає послідовність технологічних операцій проектування;

критеріїв і правил, що використовуються для оцінки результатів виконання технологічних операцій;

графічних і текстових засобів (нотацій), які використовуються для опису проектованої системи.

В основі технології проектування лежить **технологічний процес**, що визначає дії, їхню послідовність, склад виконавців, засоби і ресурси, необхідні для виконання цих дій. Технологічний процес проектування ІС у цілому поділяється на сукупність послідовно-паралельних, зв'язаних і супідрядних ланцюжків дій, кожен із яких може мати свій предмет. Дії, що виконуються в процесі проектування ІС, можуть бути визначені як неподільні **технологічні операції** (рис. 3) або як **підпроцеси технологічних операцій**.

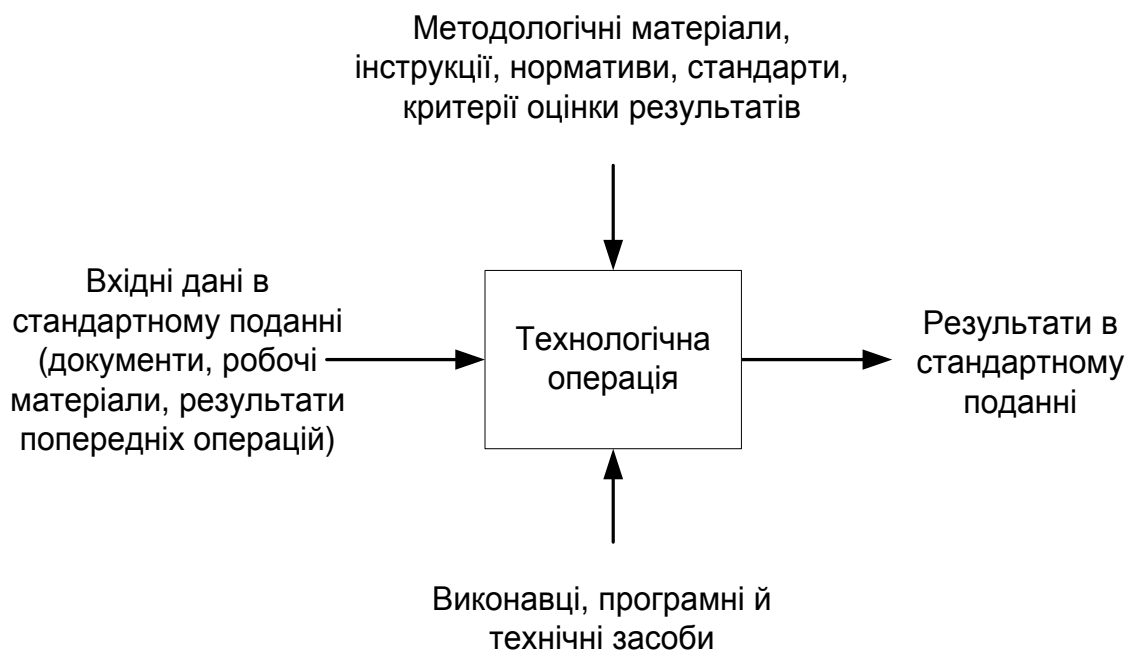


Рис. 3. Технологічна операція проектування

Усі технологічні операції можна розподілити на два основних види:

проектувальні, які формують або модифікують результати проектування;

оцінні, які виробляють за встановленими критеріями оцінки результатів проектування.

Таким чином, *технологія проектування* задається регламентованою послідовністю технологічних операцій, виконуваних у процесі створення проекту на основі того чи іншого методу, в результаті чого стало б зрозумілим не тільки те, що повинно бути зроблене для створення проекту, але й те, ким і в якій послідовності це повинно бути зроблене.

Технологічні інструкції є основною складовою технології проектування. Вони складаються з опису послідовності технологічних операцій проектування, умов, залежно від яких виконується та або інша технологічна операція, та опису самих операцій.

Предметом будь-якої обраної технології проектування повинне слугувати відображення взаємозалежних процесів проектування на всіх стадіях життєвого циклу ІС.

До основних вимог, які висувуються до обраної технології проектування, відносяться наступні:

створений за допомогою цієї технології проект повинен відповідати вимогам замовника;

обрана технологія повинна максимально відбивати всі етапи циклу життя проекту;

обрана технологія повинна забезпечувати мінімальні трудові та вартісні витрати на проектування й супровід проекту;

технологія повинна бути основою зв'язку між проектуванням і супроводом проекту;

технологія повинна сприяти зростанню продуктивності праці проектувальника;

технологія повинна забезпечувати надійність процесу проектування й експлуатації проекту;

технологія повинна сприяти простому веденню проектної документації;

технологія повинна забезпечити можливість виконання великих проектів у вигляді підсистем, тобто можливість декомпозиції проекту на складові частини, що розробляються групами виконавців обмеженої чисельності з наступною інтеграцією складових частин;

технологія повинна забезпечити можливість ведення робіт з проектування окремих підсистем невеликими групами (3 – 7 осіб), що

обумовлено принципами управління колективом і підвищенням продуктивності за рахунок мінімізації зовнішніх зв'язків;

технологія повинна забезпечити мінімальний час отримання працездатної ІС за рахунок реалізації за більш короткі строки окремих підсистем меншою кількістю розробників;

технологія повинна передбачати можливість управління конфігурацією проекту, ведення версій проекту і його складових, можливість автоматичного випуску проектної документації і синхронізацію її версій з версіями проекту;

технологія повинна забезпечувати незалежність виконуваних проектних рішень від засобів реалізації ІС – СУБД, ОС, мов і систем програмування;

технологія повинна бути підтримана комплексом так званих CASE-засобів, що забезпечують автоматизацію процесів, виконуваних на всіх стадіях процесу створення ІС.

Застосування будь-якої технології проектування конкретного проекту для конкретної організації неможливе без вироблення ряду стандартів, правил і угод, яких повинні дотримуватися усі учасники проекту. До них відносяться наступні стандарти:

проектування;

оформлення проектної документації;

призначеного для користувача інтерфейсу.

Стандарт проектування повинен встановлювати:

набір необхідних діаграм (моделей) на кожній стадії проектування і ступінь їх деталізації;

правила фіксації проектних рішень на моделях, зокрема правила присвоєння імен об'єктам (включаючи угоди з термінології), набір атрибутів для всіх об'єктів і правила їх заповнення на кожній стадії, правила оформлення діаграм, включаючи вимоги до форми та розмірів об'єктів і т. д.;

вимоги до конфігурації робочих місць розробників, включаючи настройки операційної системи, настройки CASE-засобів, спільні настройки проекту і т. д.;

механізм забезпечення спільної роботи над проектом, зокрема: правила інтеграції підсистем проекту, правила підтримки проекту в однаковому для всіх розробників проекту стані (регламент обміну проектною інформацією, механізм фіксації спільних об'єктів і т. д.), правила перевірки проектних рішень на несуперечливість і т. д.

Стандарт оформлення проектної документації повинен встановлювати:

комплектність, склад і структуру документації на кожній стадії проектування;

вимоги до оформлення документації, включаючи вимоги до змісту розділів, підрозділів, пунктів, таблиць і т. д.;

правила підготовки, розгляду, узгодження і затвердження документації із зазначенням граничних термінів для кожної стадії;

вимоги до настройки видавничої системи, використовуваної як вбудований засіб підготовки документації;

вимоги до настройки CASE-засобів для забезпечення підготовки документації відповідно до встановлених вимог.

Стандарт інтерфейсу користувача повинен встановлювати:

правила оформлення екранів (шрифти і колірну палітру), склад і розташування вікон та елементів управління;

правила використання клавіатури й мишки;

правила оформлення текстів допомоги;

перелік стандартних повідомлень;

правила обробки реакції користувача.

8.2. Формалізація технології проектування ІС

Складність, високі витрати і трудомісткість процесу проектування ІС протягом усього життєвого циклу викликають необхідність, з одного боку, вибору адекватного об'єкту технології проектування, з іншого боку, наявності ефективного інструмента управління процесом її застосування. З цього погляду виникає потреба в побудові такої формалізованої моделі технології проектування, коли на її основі можна було б оцінити необхідність і можливість застосування певної технології проектування з урахуванням сформульованих вимог до ІС і виділених ресурсів на об'єкті, а надалі – контролювати хід і результати проектування.

Відомі методи сіткового планування та управління проектами вирішують тільки одну частину поставленої проблеми: відбивають послідовність технологічних операцій з часовими і трудовими характеристиками. При цьому не розкривається повною мірою змістовна сторона процесу проектування, необхідна спочатку для розуміння сутності й оцінки ефективності технології проектування, а потім для використання як інструкційного матеріалу у безпосередній роботі проектувальників.

Найбільшою мірою завданню формалізації технології проектування ІС відповідає апарат технологічних мереж проектування. Основою формалізації технології проектування ІС є формальне визначення

технологічної операції (ТО) проектування в наступному вигляді:

$$TO = \langle V, W, П, R, S \rangle,$$

де – V – вхід;

W – вихід;

П – перетворювач;

R – ресурси;

S – засоби.

Графічна інтерпретація технологічної операції зображена на рис. 4. Технологічні операції графічно подаються у вигляді блоків-прямокутників, усередині яких даються найменування ТО, перелік використовуваних засобів проектування і посилання на використовувані ресурси. Входи і виходи ТО подаються ідентифікаторами всередині кружечків, від яких і до яких йдуть стрілки, що позначають вхідні і вихідні потоки.

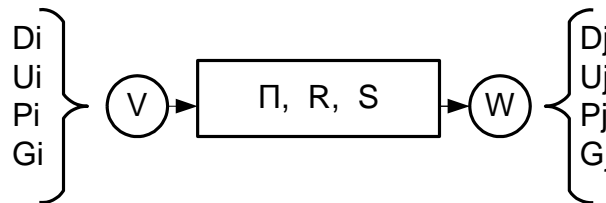


Рис. 4. Графічна інтерпретація технологічної операції

Слід детальніше розглянути компоненти формального визначення ТО.

Як компоненти входу і виходу використовуються множини документів D, параметрів P, програм G, універсальних множин (універсумів) U. Для будь-яких компонентів входу і виходу повинні бути задані форми їх подання у вигляді твердої копії чи в електронному вигляді.

Документ D – це описувач множини взаємозалежних фактів. За допомогою документів описуються об'єкти матеріальних та інформаційних потоків, організаційної структури, технічних засобів, необхідні для проектування і впровадження ІС. Документи визначають або вихідні дані проектування, або кінцеві результати проектування для реалізації нової інформаційної системи, або проміжні результати, які використовуються тимчасово для виконання наступних ТО. Кінцеві документи одночасно можуть бути і проміжними. Кінцеві документи повинні бути оформлені від-повідно до стандартів подання проектної документації.

Параметр P – це описувач одного факту. У принципі параметр

розглядається як окремий випадок документа. Виділення параметрів зі складу документів підкреслює значимість окремих фактів у процесі проектування ІС. Параметри виступають, як правило, у ролі обмежень або умов процесу проектування, наприклад, обсягу фінансування, терміну розробки, форми підприємства і т. д. Параметри можуть варіюватися з позиції аналізу впливу їх значень на результат проектування ІС.

Програма G – окремий випадок документа, що наводить опис алгоритму вирішення завдання, яке зазнає змін в міру зміни життєвого циклу ІС: від специфікації програми до машинного коду.

Універсум U – це кінцева і повна множина фактів (документів) одного типу. Звичайно за допомогою універсуму описується множина альтернатив, вибір з якої конкретного екземпляра визначає характер наступних проектних рішень. У якості універсумів можуть розглядатися множини параметризованих описів технічних засобів, програмних засобів (операційних систем, СУБД, ППП і т. д.), технологій проектування тощо.

Перетворювач P – це певна методика, формалізований алгоритм чи машинний алгоритм перетворення входу технологічної операції в її вихід. Відповідно використовуються ручні, автоматизовані й автоматичні методи реалізації перетворювачів. Для формалізації перетворювачів використовуються математичні моделі, евристичні правила, блок-схеми, псевдокоди.

Ресурси R – набір людських, комп'ютерних, часових і фінансових засобів, що дозволяють виконати технологічну операцію. Причому проектувальники можуть бути фахівцями різної кваліфікації. Наявність тих чи інших ресурсів істотно позначається на характері застосовуваної технології проектування. Наприклад, виділення мережних комп'ютерних ресурсів дозволяє здійснювати колективну розробку ІС різними групами проектувальників з паралельним виконанням технологічних операцій.

Засоби проектування S – це спеціальний вид ресурсу, що включає методичні і програмні засоби виконання технологічної операції. Якщо перетворювач є ручним, то засіб проектування становить методику виконання роботи й в описі ТО дається посилання на відповідний паперовий чи електронний документ. Якщо перетворювач є автоматизованим чи автоматичним, в описі ТО вказується посилання на назву й опис програм-ного засобу, а також посібник з його експлуатації, причому для автоматизованих перетворювачів посібник з експлуатації більшою мірою повинен бути орієнтований на методику роботи

проектувальника за допомогою даного програмного засобу.

8.3. Технологічна мережа проектування

На основі окремих технологічних операцій будується **технологічна мережа проектування (ТМП)**, під якою мається на увазі взаємозалежна за входами й виходами послідовність технологічних операцій проектування, виконання яких приводить до досягнення необхідного результату – створення проекту ІС. На ТМП технологічні операції графічно пов'язуються за допомогою спільних входів і виходів, коли вихід однієї ТО є входом іншої ТО.

Технологічні мережі проектування можуть будуватися з різним ступенем деталізації. Найбільш деталізована ТМП, у якій кожна технологічна операція є ручною, називається **канонічною**. Канонічна ТМП – найбільш придатна для проектувальників-виконавців, для яких ТМП є посібником з проектування ІС. Разом з тим канонічна ТМП усього проекту рідко використовується в повному обсязі, скоріше, різні категорії проектувальників-виконавців користуються фрагментами канонічної мережі, які стосуються їх компетенції.

Для збільшення ТМП застосовуються **технологічні операції-агрегати**, яким відповідають фрагменти канонічної ТМП. Наприклад, ТО "Проектування схеми бази даних" декомпозиується на ряд взаємозалежних ТО: "Нормалізація таблиць", "Установлення зв'язків" і т. д.

Для різних категорій учасників і розробників проекту ІС потрібен різний ступінь агрегації-деталізації ТМП. Найменш деталізована ТМП потрібна замовникам, для яких вона становить набір взаємозалежних технологічних етапів із входами, які відповідають наданій розробникам інформації, і виходами, що відповідають одержуваним проектним документам. Для керівників проектів технологічні операції, як правило, відповідають календарним роботам з чіткими термінами задачі й документальними результатами. У принципі для цих категорій користувачів ТМП може бути перетворена в традиційний сітковий графік. На цьому рівні подання ТМП можуть не вказуватися окремі ресурси чи засоби проектування.

Для проектувальників-виконавців, які взаємодіють, дуже важливе відображення в ТМП зв'язків за входом-виходом, оскільки для якісного виконання будь-якої технологічної операції необхідне точне виконання вимог за входом, що відповідає виходу іншої ТО. Для конкретного проектувальника-виконавця стосовно до його компетенції технологічна

операція-агрегат завжди може бути розкрита у вигляді фрагмента канонічної мережі.

У процесі використання засобу автоматизованого проектування проектувальник-виконавець може застосовувати технологічні операції-агрегати, що поєднують фрагменти канонічної ТМП. Для таких ТО обов'язково задається посилання на використовуваний засіб проектування. Причому якщо засіб проектування є комплексним, то вказується конкретний компонент (функція, модуль, опція і т. д.) або компоненти цього засобу.

Технологічні мережі проектування можуть мати варіанти побудови залежно від певних умов. Наприклад, ТМП проектування друкованих форм залежить від засобу проектування, вибір якого, у свою чергу, визначається складністю форм. Для правильного вибору засобу проектування з універсуму вводиться спеціальна технологічна операція, що зі-ставляє параметри вимог (наприклад, проста чи складна форма; кількість ступенів підсумків форми; кількість файлів бази даних, на підставі яких будується форма та ін.) з аналогічними параметрами засобу проектування. Залежно від обраного засобу проектування далі вибирається конкретна гілка ТМП. Наприклад, якщо в універсуму засобів проектування є тільки генератор звітів, що працює з одним файлом, то в технологічну мережу буде потрібно ввести технологічну операцію проектування вихідного файла. Якщо жоден із засобів проектування не підходить, то проектування здійснюється відповідно до канонічної мережі проектування.

8.4. Методи і засоби проектування

Основу технології проектування ІС складає **методологія проектування**, яка визначає сутність, основні відмінні технологічні особливості проектування. Методологія проектування передбачає наявність певної концепції, принципів проектування реалізованих набором методів проектування, які, у свої чергу, повинні підтримуватися певними засобами проектування.

Методи проектування – це різні способи створення ІС, що підтримуються відповідними засобами проектування. Для організації процесу проектування й автоматизації виконання проектних робіт застосовуються різні засоби проектування.

Методи проектування ІС можна класифікувати за ступенем використання засобів автоматизації, типових проектних рішень, адаптивності до передбачуваних змін (рис. 5).

За **ступенем автоматизації** методи проектування розподіляються

на методи:

ручного проектування, за якого проектування компонентів ІС здійснюється без використання спеціальних інструментальних програмних засобів, а програмування – алгоритмічними мовами;

комп'ютерного проектування, за якого генерація або конфігурація (настроювання) проектних рішень виконується на основі використання спеціальних інструментальних програмних засобів.

За **ступенем використання типових проектних рішень** розрізняють наступні методи проектування:

оригінального (індивідуального) проектування, коли проектні рішення розробляються "з нуля" відповідно до вимог до ІС;

типового проектування, що передбачає конфігурацію ІС з готових типових проектних рішень (програмних модулів).

Оригінальне проектування ІС характеризується тим, що всі види проектних робіт орієнтовані на створення індивідуальних для кожного об'єкта проектів, що максимально відображає всі його особливості. При цьому можуть створюватися не тільки індивідуальні проекти, але й відповідні методики проведення проектних робіт. Тому технологію оригінального проектування використовують у тому випадку, якщо хочуть, щоб одержуваний у результаті проектування індивідуальний проект повною мірою відображав усі особливості відповідного об'єкта управління за умов невисокої вартості розробки, зрозумілості й доступності одержуваного рішення замовнику.

До числа обмежень з використання оригінального проектування можна віднести низький ступінь автоматизації проектних робіт, тривалі терміни розробки, низьку якість документування, відсутність наступності в проектних рішеннях.

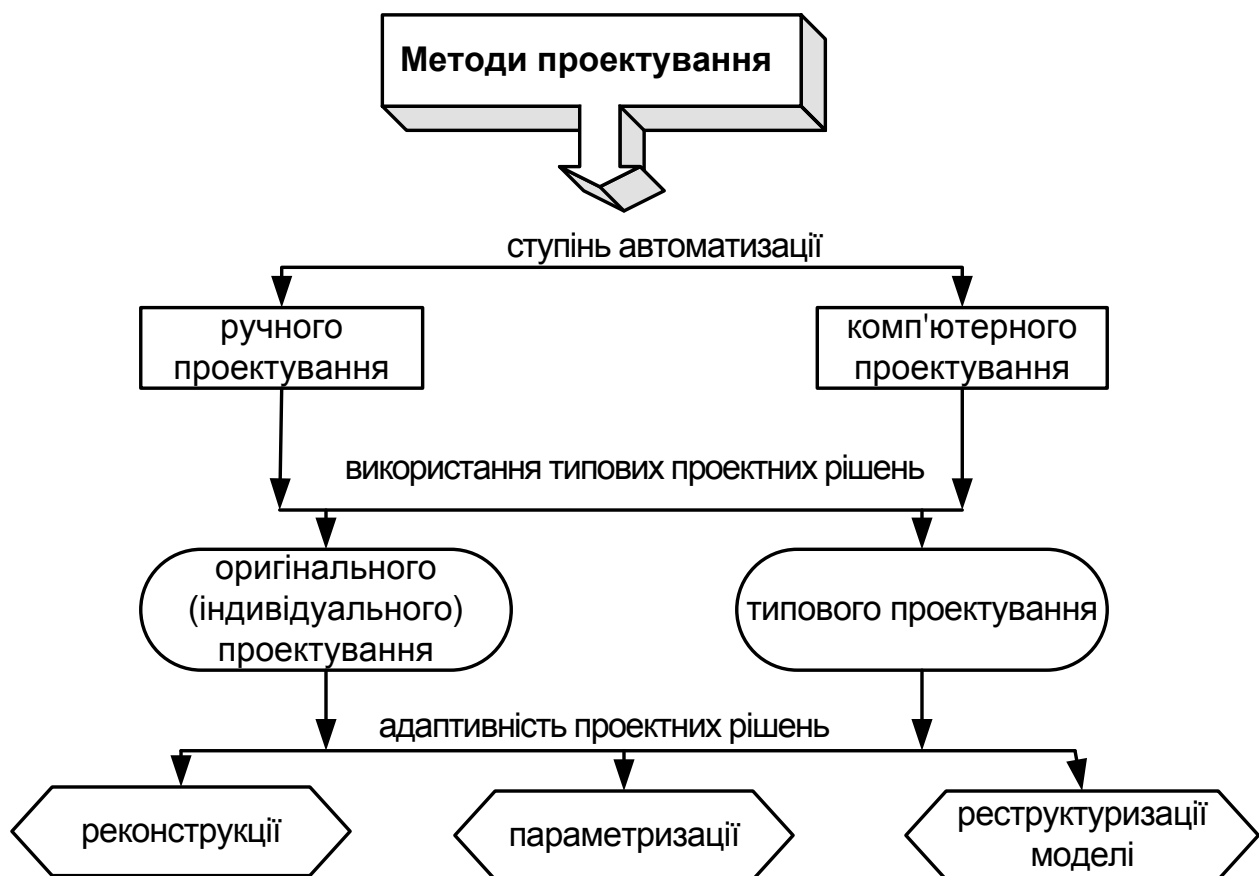


Рис. 5. Методи проектування

Типове проектування виконується на основі досвіду, отриманого під час розробки індивідуальних проектів. Типові проекти як узагальнення досвіду для деяких груп організаційно-економічних систем чи видів робіт у кожному конкретному випадку пов'язані з безліччю специфічних особливостей і розрізняються за ступенем охоплення функцій управління, виконуваними роботами і розроблюваною проектною документацією.

За **ступенем адаптивності проектних рішень** методи проектування поділяються на методи:

реконструкції, коли адаптація проектних рішень виконується шляхом переробки відповідних компонентів (перепрограмування програмних модулів);

параметризації, коли проектні рішення настраюються (перегенеруються) відповідно до змінюваних параметрів;

реструктуризації моделі, коли змінюється модель проблемної області, на основі якої автоматично перегенеруються проектні рішення.

Поєднання різних ознак класифікації методів проектування обумовлює характер використовуваної технології проектування ІС, серед яких виділяються два основних види технологій: **канонічна** й **індустріальна** (табл. 1). Індустріальна технологія проектування, у свою

чергу, розбивається на автоматизоване (використання CASE-технологій) і типове (параметрично-орієнтоване чи модельно-орієнтоване) проектування. Використання індустріальних технологій проектування не виключає використання в окремих випадках канонічної технології.

Таблиця 1

Характеристика технологій проектування

Технологія проектування	Рівень автоматизації	Рівень типізації
Канонічне проектування	Ручне проектування	Оригінальне проектування
Індустріальне проектування	Комп'ютерне проектування	Оригінальне проектування Типове проектування

Для конкретних видів технологій проектування властиве застосування певних засобів розробки ІС, що підтримують виконання як окремих проектних робіт, етапів, так і їхніх сукупностей. Тому перед розробниками ІС, як правило, постає завдання вибору засобів проектування, що за своїми характеристиками найбільшою мірою відповідають вимогам конкретного підприємства.

Засоби проектування повинні бути:

інваріантними до об'єкта проектування;

охоплювати в сукупності всі етапи життєвого циклу ІС:

технічно, програмно й інформаційно сумісними;

простими в освоєнні та застосуванні;

економічно доцільними.

Засоби проектування ІС можна розподілити на ручні й комп'ютерні (рис. 6).

Ручні засоби проектування застосовуються на всіх стадіях і етапах проектування ІС. Як правило, це засоби організаційно-методичного забезпечення операцій проектування й, у першу чергу, різні стандарти, що регламентують процес проектування систем. Сюди ж відноситься єдина система класифікації і кодування інформації, уніфікована система документації, моделі опису й аналізу потоків інформації і т. д.

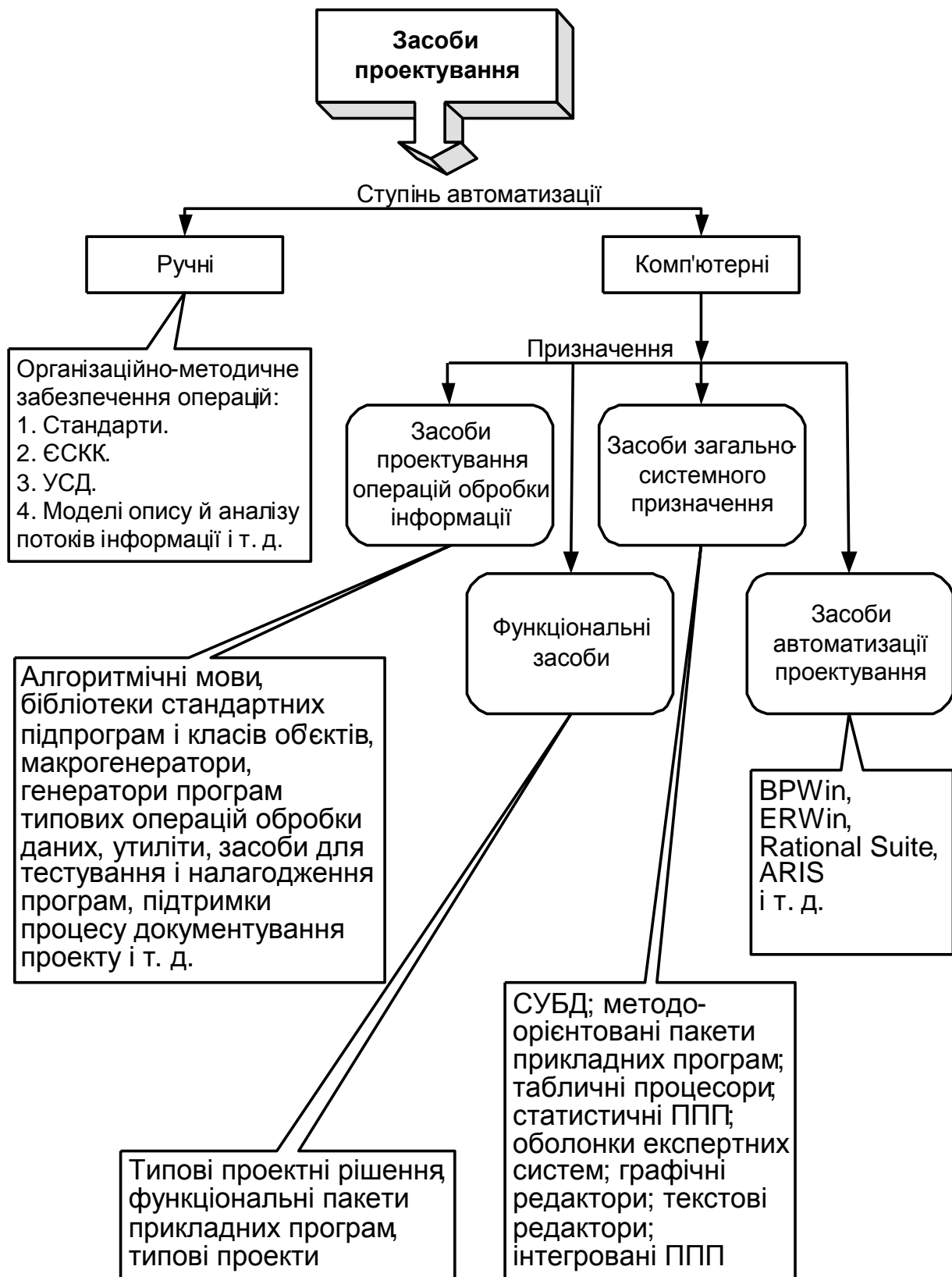


Рис. 6. Засоби проектування

Комп'ютерні засоби проектування можуть застосовуватися як на окремих, так і на всіх стадіях та етапах процесу проектування ІС і відповідно підтримують розробку елементів проекту системи, розділів

проекту системи, проекту системи в цілому. Усю множину комп'ютерних засобів проектування залежно від їх призначення поділяють таким чином:

- засоби проектування операцій обробки інформації;
- засоби загальносистемного призначення;
- функціональні засоби проектування;
- засоби автоматизації проектування.

До засобів **проектування операцій обробки інформації** належать алгоритмічні мови, бібліотеки стандартних підпрограм і класів об'єктів, макрогенератори, генератори програм типових операцій обробки даних тощо, а також засоби розширення функцій операційних систем (утиліти). Сюди включаються також такі найпростіші інструментальні засоби проектування, як засоби для тестування і налагодження програм, підтримки процесу документування проекту і т. д. Особливість останніх програм полягає в тому, що за їх допомогою підвищується продуктивність праці проектувальників, але не розробляється закінчене проектне рішення. Таким чином, засоби даного підкласу підтримують окремі операції проектування ІС і можуть застосовуватися незалежно один від одного.

До засобів **загальносистемного призначення** відносяться:

- системи управління базами даних;
- методо-орієнтовані пакети прикладних програм (вирішення завдань дискретного програмування, математичної статистики і т. д.);
- табличні процесори;
- статистичні ППП;
- оболонки експертних систем;
- графічні редактори;
- текстові редактори;
- інтегровані ППП (інтерактивне середовище із вбудованими діалоговими можливостями, що дозволяє інтегрувати перераховані вище програмні засоби).

Для перерахованих засобів проектування характерне їх використання для розробки технологічних підсистем ІС: введення інформації, організації збереження й доступу до даних, обчислень, аналізу і відображення даних, ухвалення рішень.

Функціональні засоби проектування спрямовані на розробку автоматизованих систем, які реалізують функції, комплекси завдань і завдання управління. Різноманітність предметних областей породжує різноманітність даних засобів, орієнтованих на:

- тип організаційної системи** (промислова, непромислова сфери);
- рівень управління** (наприклад, підприємство, цех, відділ, ділянка,

робоче місце);

функцію управління (планування, облік і т. д.).

До функціональних засобів проектування систем обробки інформації відносяться:

типові проектні рішення;

функціональні пакети прикладних програм;

типові проекти.

Засоби автоматизації проектування (CASE-засоби) підтримують проектування на стадіях і етапах створення ІС. Термін CASE (Computer Aided System/Software Engineering) тлумачиться досить широко – від автоматизації розробки програмного забезпечення до розробки складних ІС в цілому. Сучасні CASE-засоби, у свою чергу, класифікуються в основному за двома ознаками:

за охоплюваними етапами процесу розробки ІС;

за ступенем інтегрованості:

окремі локальні засоби (tools),

набір неінтегрованих засобів, що охоплюють більшість етапів розробки ІС (toolkit);

цілком інтегровані засоби, пов'язані загальною базою проектних даних репозиторієм (workbench).

Контрольні запитання

1. Сформулюйте мету проектування ІС.
2. Дайте визначення поняття "проект ІС".
3. Що розуміють під проектуванням ІС?
4. Дайте визначення основних компонентів проектування: "об'єкт", "суб'єкт", "предметна область"?
5. Які компоненти включає технологія проектування ІС?
6. Що таке технологічний процес проектування ІС?
7. Які вимоги висуваються до технології проектування ІС?
8. Що є технологічною операцією проектування ІС? Визначте види технологічних операцій проектування. Наведіть приклади.
9. Дайте формальне визначення технологічної операції.
10. Охарактеризуйте компоненти формального визначення технологічної операції.
11. Що таке технологічна мережа проектування?
12. Що таке канонічна технологічна мережа проектування?

13. Для чого застосовуються технологічні операції-агрегати? Наведіть їх приклади.
14. Що таке методологія проектування ІС?
15. Наведіть класифікацію методів проектування ІС.
16. Які ознаки характеризують канонічне проектування ІС?
17. Які ознаки характеризують автоматизоване проектування ІС?
18. Які ознаки характеризують типове проектування ІС?
19. Що таке індустріальне проектування ІС?
20. Наведіть класифікацію засобів проектування ІС, супроводжуючи її прикладами.

9. Програмна інженерія

9.1. Поняття життєвого циклу програмного продукту

9.1.1. Життєвий цикл промислового виробу

На початку 70-х років минулого століття виникла криза програмування, суть якої полягала в наступному. Зростання складності й масштабів проектів призвело до того, що в процесі розробки програмних проектів:

порушувалися терміни розробки;
перевищувалися заплановані обсяги фінансування;
отримані результати не відповідали вимогам замовника;
супровід програмних продуктів вимагав більше коштів, ніж сама розробка;

частина проектів взагалі не доводилися до завершення.

Тому виникла необхідність в радикальному вдосконаленні принципів і методів розробки програмних продуктів (ПП) з урахуванням їх модернізації та супроводу. Виходом із цієї ситуації було використання досвіду промислового проектування складних проектів, так званої **промислової інженерії**.

Методологічну основу промислової інженерії складає поняття **життєвого циклу виробу (продукту)** як сукупності всіх дій, які треба виконати впродовж всього "життя" виробу. *Суть життєвого циклу полягає у взаємопов'язаній сукупності всіх цих дій.*

Життєвий цикл промислового виробу становить послідовність етапів: проектування, виготовлення зразка, організації виробництва, серійного виробництва, експлуатації, ремонту, виведення з експлуатації,

які складаються з технологічних процесів, дій і операцій. Організація промислового виробництва з позиції життєвого циклу дозволяє розглядати всі його етапи у взаємозв'язку, що дає можливість скоротити часові, вартісні й трудові витрати на виробництво і супровід продуктів.

Промислові вироби характеризуються початковим (Сп), цільовим (Сц) і кінцевим (Ск) станами.

Початковим станом виробу є момент виникнення задуму (ідеї) або початку фінансування процесу його проектування (створення).

Цільовий стан пов'язується з моментом початку функціонування, тобто виконання об'єктом свого призначення, задоволення потреб середовища, для якого створюється об'єкт.

Кінцевий стан пов'язується з моментом його елімінації, тобто припинення діяльності у зв'язку з фізичним чи моральним старінням або зміною, перетворенням у якісно новий об'єкт.

Упорядкована сукупність змін стану виробу між початковим і кінцевим станами називається **життєвим циклом (ЖЦ)** виробу:

$$Сп \rightarrow Сц \rightarrow Ск.$$

Уся сукупність змін стану виробу між Сп і Сц відповідає **творчій стадії**, під час якої здійснюються проектування виробу і його уречевлення (матеріальне, енергетичне, інформаційне), тобто виготовлення продукту. Сукупність змін стану об'єкта між Сц і Ск відповідає **стадії експлуатації** продукту.

Кінцевих цілей продукту може бути декілька, тому в загальному випадку цільовий стан можна подати у вигляді сукупності цілей:

$$Сц = \{Сц_1, Сц_2, \dots, Сц_n\}.$$

Одна чи декілька цілей з огляду на низку причин може не виконуватися, і система може перейти в один із кінцевих станів:

$$Ск = \{Ск_1, Ск_2, \dots, Ск_n\}.$$

9.1.2. Визначення програмної інженерії

Програмна інженерія (або технологія програмування) як певний напрям виникла і формувалася під тиском зростання вартості

створюваного програмного забезпечення. *Головна мета програмної інженерії – скорочення вартості й термінів розробки програм.*

Термін "**програмна інженерія**" ("software engineering") з'явився в жовтні 1968 року на конференції підкомітету НАТО з науки і техніки, що проходила в Германії. На ній були присутні розробники програмного забезпечення (ПЗ) з 11 країн. Розглядалися проблеми проектування, розробки, поширення й підтримки програм. Термін "програмна інженерія" був поданий як певна дисципліна, яку треба створювати і якою треба керуватися у вирішенні проблем, пов'язаних із розробкою ПЗ.

Програмна інженерія – це інженерна дисципліна, яка пов'язана зі всіма аспектами виробництва ПЗ: від початкових стадій створення специфікації до підтримки системи після здачі в експлуатацію. Розглянемо детальніше це визначення.

В інженерній практиці для вирішення нових завдань шукають нові методи або засоби, застосовують їх і відповідають за результат, оскільки метод або засіб ще не перевірені. Набір таких інженерних методів або способів, теоретично можливо не обґрунтованих, але неодноразово підтверджених на практиці, відіграє велику практичну роль. У програмній інженерії ці методи отримали назву **кращих практик (best practices)**.

Створення програмних продуктів пов'язане з обмеженнями ресурсів: часових, фінансових і організаційних (устаткування, техніки, людей), тобто програмний продукт повинен бути створений у встановлені терміни, в рамках виділених коштів, устаткування і людей. Це стосується як створення замовлених продуктів, що передбачається умовами договору, так і створення коробкових продуктів, що пов'язано з ринковою конкуренцією продуктів.

Програмна інженерія займається не тільки технічними питаннями виробництва ПЗ (специфікацією вимог, проектуванням, кодуванням і т. д.), але й управлінням програмними проектами, включаючи питання планування, фінансування, управління колективом тощо. Крім того, завданням програмної інженерії є розробка засобів, методів і теорій для підтримки процесу виробництва ПЗ.

Програмні інженери застосовують систематичні й організовані підходи до роботи для досягнення максимальної ефективності і якості ПЗ. Їх

завдання полягає в адаптації існуючих методів і підходів до вирішення своєї конкретної проблеми.

Слід зазначити, що частка провальних проектів у програмній інженерії більша порівняно з іншими інженеріями. Тому ключовим питанням програмної інженерії є можливість застосування досвіду інших інженерій. Доцільно розглянути основні відмінності програмних продуктів від традиційних промислових продуктів.

Перша відмінність. Комп'ютерна програма, на відміну від об'єктів інших інженерій, не матеріальний об'єкт. Пристрій пам'яті будь-якого типу – це лише носій програми.

Слід зазначити, що життєвий цикл продукту будь-якої інженерії в спрощеному вигляді включає фази: проектування, створення зразка, випробування, виробництва, експлуатації. Тому можна перерахувати основні відмінності комп'ютерної програми від традиційних промислових продуктів за фазами життєвого циклу:

фаза виробництва полягає в копіюванні зразка на інші носії, вартість фази дуже низька;

фаза створення зразка практично відсутня, оскільки кодування можна вважати елементом проектування.

Звідси випливає, що вартість програми – це вартість лише її проектування. У процесі проектування коробкових продуктів вона розкладається на тиражовані копії, за рахунок чого вартість однієї копії стає низькою. Вартість замовлених, масово не копіюваних продуктів залишається дуже високою.

Друга відмінність. Програма є штучним об'єктом. Тобто для програми немає об'єктивних законів, яким би підкорялася її поведінка. Так, наприклад, для проектування різних споруд існують об'єктивні закони, яким підкоряються об'єкти будівництва. Тому можна перевірити інженерні рішення на відповідність цим законам і тим самим забезпечити успіх проекту, що не може зробити програмний інженер.

Прямим наслідком відсутності можливості "теоретичного" контролю проекту є те, що *тестування продукту – це єдиний спосіб переконатися в його якості*. Саме тому вартість тестування складає істотну частину вартості ПЗ (рис. 7). Так на практиці, формулювання вимог і умов розробки проекту (специфікація вимог) становить 15% від

вартості проекту, розробка й верифікація проекту (проекткування) – 25%, кодування та тестування компонент програми (розробка) – 20%, а об'єднання і складальне тестування продукту (інтеграція і тестування) – 40%. (Передбачається що роботи супроводу включені таким чином: виправлення помилок – до продовження тестування, а внесення змін – до нового проекту.)

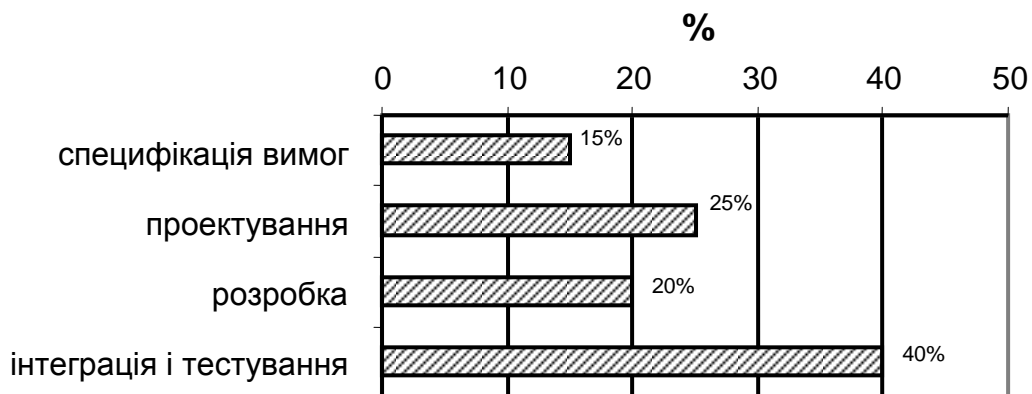


Рис. 7. Типовий розподіл вартості між основними етапами

Програмна інженерія пройшла декілька етапів розвитку, в процесі яких були сформульовані фундаментальні принципи і методи розробки програмних продуктів. **Основний принцип** програмної інженерії полягає в тому, що програми створюються в результаті виконання дій декількох взаємопов'язаних етапів (аналізу вимог, проектування, реалізації, впровадження, супроводу), складових життєвого циклу програмного продукту. Незважаючи на те що програмна інженерія ще досить молода дисципліна, на сьогодні в ній вже накопичений певний досвід. За умови правильного застосування цього досвіду можна створювати успішні проекти.

9.1.3. Особливості життєвого циклу програмного продукту

Основу програмної інженерії складає поняття життєвого циклу програмного продукту. Під **життєвим циклом програмного продукту** (SLC – Software Lifetime Cycle) розуміють послідовність фаз (стадій, етапів) які необхідно виконати в процесі створення й експлуатації ПЗ.

Концепція життєвого циклу програмного продукту була запропонована в 1968 р. в Лондоні на зустрічі керівників проектів з розробки програмного

забезпечення. Потім у 1970 р. У. У. Ройсом були ідентифіковані окремі стадії в типовому циклі програмного продукту. Контроль за виконанням виділених стадій повинен був привести до підвищення якості ПЗ і скорочення вартості розробки.

Суть змісту життєвого циклу (ЖЦ) розробки ПЗ у різних підходах однакова і зводиться до виконання наступних укрупнених стадій (фаз):

1. Системний аналіз – планування й аналіз вимог до ПЗ.
2. Розробка (системний синтез) – проектування та реалізація ПЗ.
3. Супровід ПЗ – експлуатація та модернізація ПЗ.

Важливою рисою життєвого циклу ПЗ є його повторюваність: системний аналіз – розробка – супровід – системний аналіз (рис. 8). Це відповідає уявленню про ПЗ як про динамічну систему. При першому виконанні стадії "Розробка" створюється проект ПЗ, а при повторному виконанні – здійснюється модифікація проекту для підтримки його в актуальному стані.

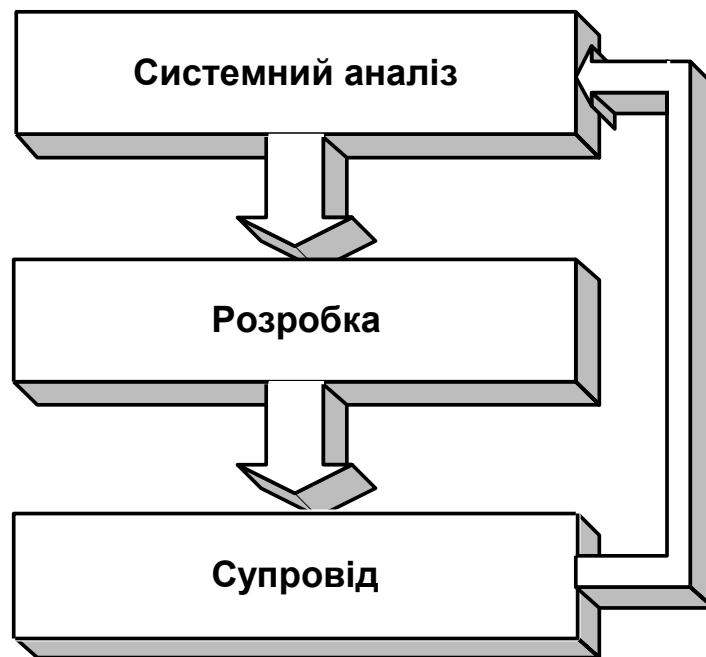


Рис. 8. Життєвий цикл проектування ПЗ

Іншою характерною рисою життєвого циклу є наявність циклів (рис. 9): перший цикл – це цикл первинного проектування ПЗ; другий цикл – цикл, що виникає після дослідного впровадження, у результаті якого з'ясовуються окремі помилки в елементах проекту; третій цикл виникає після здачі в промислову експлуатацію, коли

виявляють помилки у функціональній архітектурі системи, пов'язані з невідповідністю проекту вимогам замовника, за складом функціональних підсистем, складом завдань і зв'язками між ними;

четвертий цикл виникає в тому випадку, коли потрібна модифікація системної архітектури у зв'язку з необхідністю адаптації проекту до нових умов функціонування системи;

п'ятий цикл виникає, якщо проект системи цілком не відповідає вимогам, які висуваються до організаційно-економічної системи, через те, що здійснюється його моральне старіння і потрібне повне перепроектування системи.

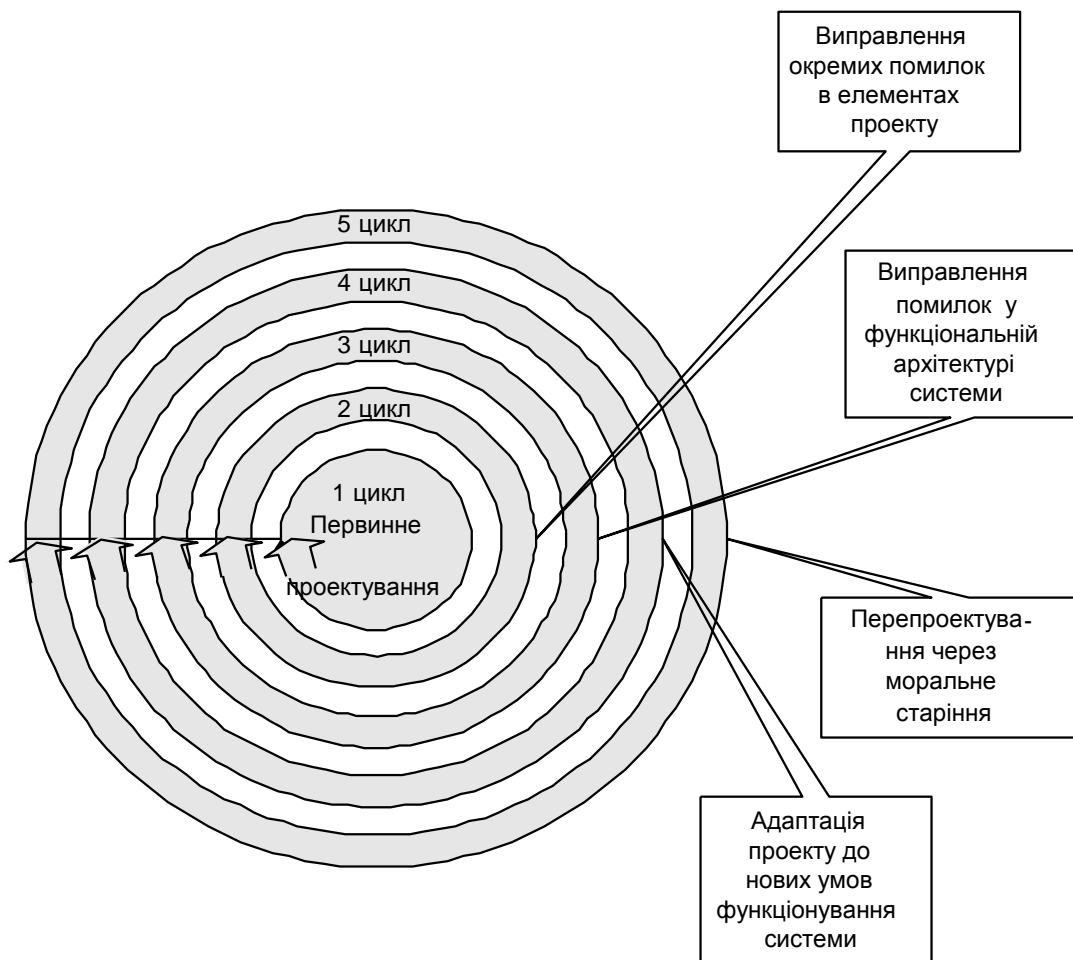


Рис. 9. Цикли в проектуванні програмного забезпечення

Щоб виключити п'ятий цикл і максимально зменшити необхідність виконання третього і четвертого циклів, необхідно виконувати проектування ПЗ на всіх етапах першого, основного циклу розробки ПЗ відповідно до вимог:

розробка ПЗ повинна бути виконана в чіткій відповідності зі сформульованими вимогами до створюваної системи;

вимоги до ПЗ повинні адекватно відповідати цілям і завданням ефективного функціонування об'єкта;

створене ПЗ повинне відповідати сформульованим вимогам на момент закінчення впровадження, а не на момент початку розробки;

упроваджене ПЗ повинне розвиватися й адаптуватися відповідно до постійно змінюваних вимог до ПЗ.

9.2. Стандарти життєвого циклу програмного забезпечення

9.2.1. Стандарти і сертифікація

Основу промислового виробництва програмних продуктів становлять процеси стандартизації та сертифікації. У процесі виробництва товарів або послуг організація застосовує певну технологію виробництва. Ця технологія повинна відповідати стандартам на товари або послуги. Використовувана організацією технологія проходить сертифікацію на відповідність цим стандартам.

Слово "*технологія*" походить від грецьких слів *techne* (мистецтво, ремесло) і *logos* (поняття, вчення). **Технологія** визначається як сукупність прийомів і способів отримання, обробки чи переробки сировини, матеріалів, напівфабрикатів або виробів, здійснюваних у різних галузях промисловості, в будівництві і т. д. Технологія становить докладний опис того, як треба виготовляти той чи інший виріб.

Слово "*стандарт*" походить від англійського слова *standard* (норма, зразок, мірило). **Стандарт** – це ухвалений компетентним органом нормативно-технічний документ, що встановлює комплекс норм, правил відносно до предмета стандартизації, або типовий зразок, еталон, модель, що беруться за основу для зіставлення з ними інших предметів. Наприклад, ДСТУ 3918-1999 (ISO/IEC 12207:1995). "Інформаційні технології. Процеси життєвого циклу програмного забезпечення" – це документ, що описує термінологію для процесів ЖЦ ПЗ, а також що визначає процеси, які використовуються для визначення, управління і вдосконалення життєвого циклу програмного забезпечення. Прикладом типових зразків може слугувати еталонна система мір і ваг. Так еталон метра зберігається в палаті мір і ваг, що знаходиться в Парижі.

Стандарт може бути розроблений на:

матеріально-технічні предмети (продукцію, еталони, зразки речовин);

норми, правила, вимоги організаційно-методичного і загально-технічного характеру.

Слово "*сертифікат*" походить від латинських слів *certum* – *правильно* і *facere* – *робити* і означає "зроблено правильно".

Для сертифікації продукту треба знати:

яким вимогам він повинен відповідати;

яким чином можна отримати вірогідні докази цієї відповідності.

Загальновизнаним способом такого доказу є заява про відповідність і сертифікація відповідності.

Заява постачальника про відповідність означає, що постачальник (виробник) під свою особисту відповідальність повідомляє про те, що його продукція відповідає вимогам конкретного нормативного документа і містить відомості, які ідентифікують виробника, виріб, стандарт, а також вказівку про особисту відповідальність виробника за зміст заяви. Заява не є гарантією відповідності стандарту. Заява відбиває готовність нести відповідальність.

Сертифікація відповідності визначається як дія третьої сторони, яка доводить, що забезпечується необхідна впевненість у тому, що належним чином ідентифікована продукція, процес або послуга відповідає конкретному стандарту або іншому нормативному документу.

Сертифікація відповідності передбачає обов'язкову участь третьої сторони і здійснюється за правилами певної процедури, яка включає обов'язкові випробування на відповідність стандарту. Сертифікація вважається основним вірогідним способом доказу відповідності продукції (процесу, послуги) заданим вимогам (стандартам).

Систему сертифікації (у загальному вигляді) складають:

центральний орган, який управляє системою, здійснює нагляд за її діяльністю і може передавати право на проведення сертифікації іншим органам; правила і порядок проведення сертифікації;

нормативні документи, на відповідність яким здійснюється сертифікація;

процедури (схеми) сертифікації;

порядок інспекційного контролю.

Системи сертифікації можуть діяти на національному, регіональному і міжнародному рівнях.

9.2.2. Види стандартів

Існуючі стандарти прийнято ділити на наступні основні види: корпоративні, галузеві, державні й міжнародні.

Корпоративні стандарти розробляються великими фірмами (корпораціями) з метою підвищення якості своєї продукції. Такі стандарти розробляються на основі власного досвіду і з урахуванням вимог світових стандартів. Корпоративні стандарти не сертифікуються, але є обов'язковими для застосування всередині корпорації. В умовах

ринкової конкуренції вони можуть мати закритий характер. У сфері IT-технологій відомі стандарти, розроблені фірмами IBM, Intel, Microsoft тощо.

Галузеві стандарти діють у межах організацій певної галузі (міністерства, наприклад, стандарти спеціальностей, яким навчають у навчаль-них закладах Міністерства освіти і науки України. Ці стандарти розробляються з урахуванням вимог світового досвіду і специфіки галузі, є, як правило, обов'язковими для галузі і підлягають сертифікації.

Державні стандарти (ГОСТ, ДСТУ) приймаються державними органами, мають силу закону. Розробляються з урахуванням світового досвіду або на основі галузевих стандартів. Можуть мати як рекомендаційний, так і обов'язковий характер, наприклад, стандарти пожежної безпеки. Для сертифікації створюються державні органи або органи ліцензування.

Міжнародні стандарти розробляються, як правило, спеціальними міжнародними організаціями на основі світового досвіду і кращих корпоративних стандартів. Мають суто рекомендаційний характер. Право сертифікації отримують організації (державні й приватні), що пройшли ліцензування в міжнародних організаціях. Основні організації-розробники міжнародних стандартів у сфері програмної інженерії наведені в табл. 2.

Таблиця 2

Організації-розробники міжнародних стандартів у сфері програмної інженерії

Скорочена назва	Назва англійською мовою	Назва українською мовою	Види стандартів
ISO	International Organization for Standardization	Міжнародна організація зі стандартизації	Міжнародні стандарти: якості – ISO 9000; процесів життєвого циклу ПЗ – ISO/IEC 12207
SEI	Software Engineering Institute	Інститут програмної інженерії	Програмна інженерія: зрілості організацій, що розробляють ПЗ, – CMM™
PMI®	Project Management Institute	Інститут управління проектами	Загальний менеджмент проектів: основи знань у галузі управління проектами – PMBOK
IEEE	Institute of Electrical and Electronics Engineers	Інститут інженерів електротехніки та електроніки	Зведення стандартів з програмної інженерії IEEE

ISO – це неурядова організація з консультативним статусом ООН. Головна її мета – розвиток стандартизації, а також споріднених напрямів діяльності у всьому світі. Міжнародна організація зі стандартизації ISO є всесвітньою федерацією національних організацій зі стандартизації (комітетів-членів ISO). Розробка міжнародних стандартів здійснюється технічними комітетами ISO. Міжнародні урядові й неурядові організації, що мають зв'язки з ISO, також беруть участь у роботах. Стандарти ISO є рекомендаційними; водночас деякі міжнародні стандарти (наприклад, з проблем охорони здоров'я, безпеки, охорони навколишнього середовища) прийняті рядом країн як обов'язкові на території даної країни. Сертифікації на відповідність своїм стандартам і контролю за їх виконанням організація не здійснює. Цим зазвичай займаються спеціально призначені державні органи реєстрації, лабораторії або аудиторські інститути. Ці процедури, зокрема, можуть виконувати на комерційній основі приватні аудитори. Проекти міжнародних стандартів приймаються технічними комітетами і розсилаються комітетам-членам на голосування. Для публікації стандартів у якості міжнародних необхідне схвалення не менше 75% комітетів-членів від числа тих, що беруть участь у голосуванні. Використання логотипу ISO на яких-небудь продуктах або в підприємствах є також незаконним, оскільки передбачає негласну сертифікацію даного продукту.

SEI – організація, яка заснована при університеті Карнегі/Меллон. Дослідження й розробки в організації проводяться на основі федерального фінансування і спонсоруються Міністерством оборони США. Основне завдання SEI – створення методик для оцінки рівня розвитку внутрішніх процесів в організації. З 1984 року організація розвиває і пропагує методики для розробки високоякісного ПЗ.

PMI – це міжнародна громадська організація, яка об'єднує професіоналів у галузі проектного менеджменту, яка була заснована в 1969 р. у США. У PMI беруть участь члени зі 125 країн світу. Основною метою PMI є просування, пропаганда, розвиток проектного менеджменту в різних країнах. Організація розробляє стандарти в галузі проектного менеджменту, а також займається підвищенням кваліфікації кадрів. PMI є провідною професійною організацією з управління проектами в наступних галузях: авіакосмічна й автомобільна промисловість, управління комерційними під-приємствами, машинобудування, фінансові операції, інформаційні технології, фармацевтика, телекомунікації та багато інших.

PMI надає всеосяжне керівництво розробкою стандартів для проектного менеджменту – стандарт з управління проектами PMBOK. PMI став першою організацією у світі, що має програму сертифікації фахівців з управління проектами – Project Management Professional (PMP).

IEEE (вимовляється як ай-тріпл-і) – організація, що об'єднує близько 400 тис. технічних фахівців із більш ніж 150 країн. До IEEE входять ряд професійних товариств. Найбільшим серед них є IEEE Computer Society, куди входять більше 100 тис. осіб. IEEE Computer Society широко відоме своєю діяльністю зі стандартизації, яку на сьогодні в рамках товариства здійснюють близько 200 робочих груп. Комп'ютерне товариство IEEE щорічно спонсорує близько ста п'ятдесяти наукових конференцій і симпозіумів, публікує більше 20 періодичних видань.

9.2.3. Стандарти процесів життєвого циклу програмного забезпечення

Розробка і практичне застосування стандартів стосовно ЖЦ ПЗ стикалися з рядом проблем:

- вкладення значних коштів на застосування стандартів, що не завжди окупалося;

- не було ясності, які процеси треба виконувати і в якому обсязі;

- до різних типів програмних систем (ІС організаційного управління, управління технологічними процесами і т. д.) висувуються різні вимоги;

- динамічність ІТ-галузі приводила до швидкого "старіння" стандартів;

- термінологічна невідповідність різних корпоративних стандартів.

Тому світовими організаціями було розроблено ряд стандартів у галузі програмної інженерії, в яких вони намагалися вирішити ці проблеми. Серед найбільш відомих стандартів програмної інженерії можна виділити наступні:

- ISO/IEC 12207** – Information Technology – Software Life Cycle Processes – Процеси життєвого циклу програмних засобів;

- ISO/IEC 15504** – Software Process Assessment – Оцінка і атестація зрілості процесів створення і супроводу ПЗ;

- SEI CMM** – Capability Maturity Model (for Software) – Модель зрілості процесів розробки програмного забезпечення;

- SEI CMMI** (Capability Maturity Model Integration) – Інтеграція моделей зрілості процесів розробки програмного забезпечення;

PMBOK – Project Management Body of Knowledge – Зведення знань з управління проектами. Стандарт PMBOK розроблений і розвивається PMI. Він містить описи 9 розділів (галузей знань) управління проектами:

- управління інтеграцією – Project Integration Management;
- управління обмеженнями – Project Scope Management;
- управління часом – Project Time Management;
- управління витратами – Project Cost Management;
- управління ризиками – Project Risk Management;
- управління персоналом – Project Personnel Management;
- управління комунікаціями – Project Communication Management;
- управління закупівлями – Project Procurement Management;
- управління якістю – Project Quality Management.

IEEE SWEBOK – IEEE Computer Society Software Engineering Body of Knowledge Software Engineering Body of Knowledge – Зведення знань з програмної інженерії – проект IEEE Computer Society. Основна ідея проекту аналогічна до PMBOK і полягає у створенні певного базового набору загальноприйнятих знань, необхідних будь-якому професійному програмістові. Він містить описи наступних 10 розділів (галузей знань) програмної інженерії:

- Software Requirements – вимоги до ПЗ;
- Software Design – проектування ПЗ;
- Software Construction – конструювання ПЗ;
- Software Testing – тестування ПЗ;
- Software Maintenance – супровід ПЗ;
- Software Configuration Management – управління конфігураціями;
- Software Engineering Management – управління IT-проектом;
- Software Engineering Process – процес програмної інженерії;
- Software Engineering Tools and Methods – методи та інструменти;
- Software Quality – якість ПЗ.

Стандарти є сумою досвіду, накопиченого експертами в інженерії ПЗ на основі великої кількості проектів, що проводилися в рамках комерційних структур США і Європи, а також у рамках військових контрактів. Велика частина стандартів створювалася як набір критеріїв для відбору постачальників програмного забезпечення для Міністерства оборони США, і це завдання вони вирішують достатньо успішно.

9.2.4. Стандарти процесів ЖЦ ПЗ, розроблені ISO

Значними кроками у вирішенні проблем стандартизації ЖЦ ПЗ стали розробка й ухвалення в 1995 р. стандарту ISO/IEC 12207 – Information Technology – Software Life Cycle Processes. У 2000 р. він був прийнятий в

Україні як ДСТУ 3918-1999 "Інформаційні технології. Процеси життєвого циклу програмного забезпечення".

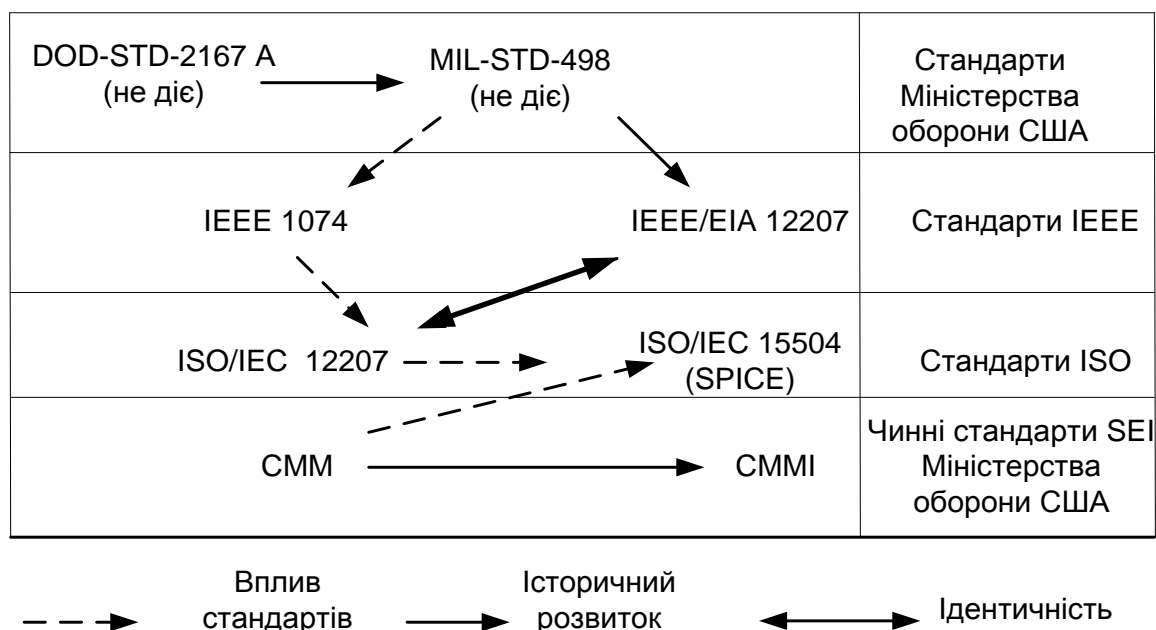


Рис. 10. Зв'язок між стандартами ЖЦ ПЗ

Стандарт ISO/IEC 12207 розроблявся з урахуванням кращого світового досвіду на основі наступних стандартів (рис. 10):

DOD-STD-2167 A "Розробка програмних засобів для систем військового призначення". Стандарт був прийнятий у 1985 р., а в 1988 р. – уточнений. Це перший формалізований і затверджений стандарт життєвого циклу для проектування програмних систем військового призначення на замовленнях Міністерства оборони США. Цим документом регламентовано 8 фаз (етапів) у процесі створення складних критичних програмних систем і близько 250 типових обов'язкових вимог до процесів і об'єктів проектування на цих етапах;

MIL-STD-498 "Розробка і документування програмного забезпечення". Стандарт прийнятий Міністерством оборони США в 1994 р. для заміни DOD-STD-2167A та ряду інших стандартів. Він призначений для застосування всіма організаціями і підприємствами, що одержують замовлення Міністерства оборони США. У 1996 р. затверджено докладне керівництво "Застосування і рекомендації до стандарту MIL-STD-498". Основну частину складають 75 підрозділів – рекомендацій щодо забезпечення й реалізації процесів ЖЦ складних критичних програмних систем високої якості та надійності, що функціонують у реальному часі;

IEEE 1074 "Процеси життєвого циклу для розвитку програмного забезпечення". Стандарт прийнятий у 1995 р. і охоплює повний життєвий цикл програмних систем, у якому виділяються шість великих базових процесів. Ці процеси деталізуються 16 приватними процесами. В останніх є ще дрібніша деталізація в сукупності на 65 процесів-робіт. Зміст кожного

окремого процесу починається з опису загальних його функцій і завдань та переліку дій – робіт під час подальшої деталізації. Для кожного процесу в стандарті наведена вхідна і результативна інформація про його виконання і короткий опис суті процесу. У стандарті увага зосереджена переважно на безпосередньому створенні ПЗ і на процесах попереднього проектування.

Основними результатами стандарту ISO 12207 є:

введення єдиної термінології з розробки й застосування ПЗ, призначеної не лише для розробників, але й для замовників, користувачів, усіх зацікавлених осіб;

опис організації ЖЦ і його структури (процесів).

Стандарт поширюється на процеси замовлення систем, програмних продуктів та послуг, процеси постачання, розробки, функціонування та супроводу програмних продуктів, включно з програмним компонентом програмно-апаратних засобів, які виконуються як самою організацією, так і поза її межами. Причому процеси, що використовуються протягом життєвого циклу програмного забезпечення, повинні бути сумісними з процесами, що використовуються протягом життєвого циклу системи.

У стандарті ISO /IEC 12207 даються визначення термінів стосовно розробки та застосування ПЗ, наведених у стандартах ISO 8402, ISO/IEC 2382-1 та ISO/IEC 2382-20 (додаток А).

Стандарт визначає організацію життєвого циклу програмного продукту як сукупність процесів, кожний із яких розбитий на дії, що складаються з окремих завдань, встановлює структуру (архітектуру) життєвого циклу програмного продукту у вигляді переліку процесів, дій і завдань (рис. 11).

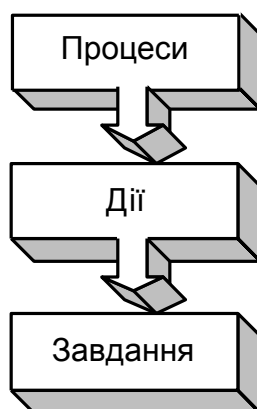


Рис. 11. Організація ЖЦ ПЗ згідно зі стандартом ДСТУ 3918-1999 (ISO/IEC 12207)

Процес – це множина взаємопов’язаних дій і ресурсів, які перетворюють входи на виходи.

Дія – це елемент процесу проекту. У дії звичайно наявна очікувана тривалість, потреба в ресурсах, вартість. Дії можуть далі розподілятися на завдання.

Відповідно до стандарту ISO/IEC 12207, процеси організації ЖЦ ПЗ діляться на три групи: основні процеси, процеси підтримки, організаційні процеси. Окремо описаний процес адаптації стандарту, що містить основні роботи, які повинні бути виконані під час адаптації даного стандарту до умов конкретного програмного проекту (табл. 3)

Таблиця 3

Процеси життєвого циклу ПЗ (ISO/IEC 12207)

Основні процеси	Процеси підтримки	Організаційні процеси	Процес адаптації
замовлення; постачання; розробка; експлуатація; супровід	документування; конфігурування; забезпечення якості; верифікація; валідація; спільний перегляд; аудит; вирішення проблем	управління; створення інфраструктури; вдосконалення; навчання	адаптація процесів до умов конкретного проекту

До **основних процесів** життєвого циклу відносяться наступні (рис. 12):

замовлення – визначає дії замовника, тобто організації, яка замовляє систему, програмний продукт або програмну послугу;

постачання – визначає дії постачальника, тобто організації, яка надає систему, програмний продукт або програмну послугу замовнику;

розробка – визначає дії розробника, тобто організації, яка визначає та проектує програмний продукт;

експлуатація – визначає дії оператора, тобто організації, що надає послуги із експлуатації комп’ютерної системи в її наявному середовищі для її користувачів;

супровід – визначає дії супроводжувача, тобто організації, яка надає послуги із супроводу програмного продукту, тобто керує внесенням змін до програмного продукту з метою підтримання його в належному та працездатному стані. Даний процес охоплює перенесення і вилучення з експлуатації програмного продукту.

До **процесів підтримки** життєвого циклу відносяться:

документування – визначає дії щодо реєстрації інформації, виробленої процесом життєвого циклу;

конфігурування – визначає дії щодо управління конфігурацією;

забезпечення якості – визначає дії щодо набуття об'єктивної впевненості в тому, що програмні продукти та процеси відповідають заданим для них вимогам та відповідають встановленим для них планам. Як методи забезпечення якості можна використовувати процеси спільного перегляду, аудиту, верифікації та валідації;

верифікація – визначає дії (замовника, постачальника або незалежного учасника) щодо проведення верифікації програмного продукту з різним ступенем глибини залежно від програмного проекту;

валідація – визначає дії замовника, постачальника, або незалежного учасника щодо проведення валідації програмного продукту, одержаного в рамках програмного проекту;

спільний перегляд – визначає дії щодо проведення оцінки стану та результатів певної дії. Цей процес може бути застосований будь-якими двома учасниками, один із яких (учасник, що здійснює перегляд) здійснює перегляд дій іншого учасника (учасника, дії якого переглядаються в рамках спільного обговорення);

аудит – визначає дії щодо визначення відповідності вимогам, планам та контракту. Цей процес може бути застосований будь-якими двома учасниками, один із яких (учасник, що перевіряє) проводить аудит програмного продукту або дій іншого учасника (учасника, якого перевіряють);

вирішення проблем – визначає дії щодо аналізу і зняття проблем (включно з невідповідностями) незалежно від їхньої природи та причин, виявлених у процесі розробки, експлуатації, супроводу чи під час виконання інших процесів.

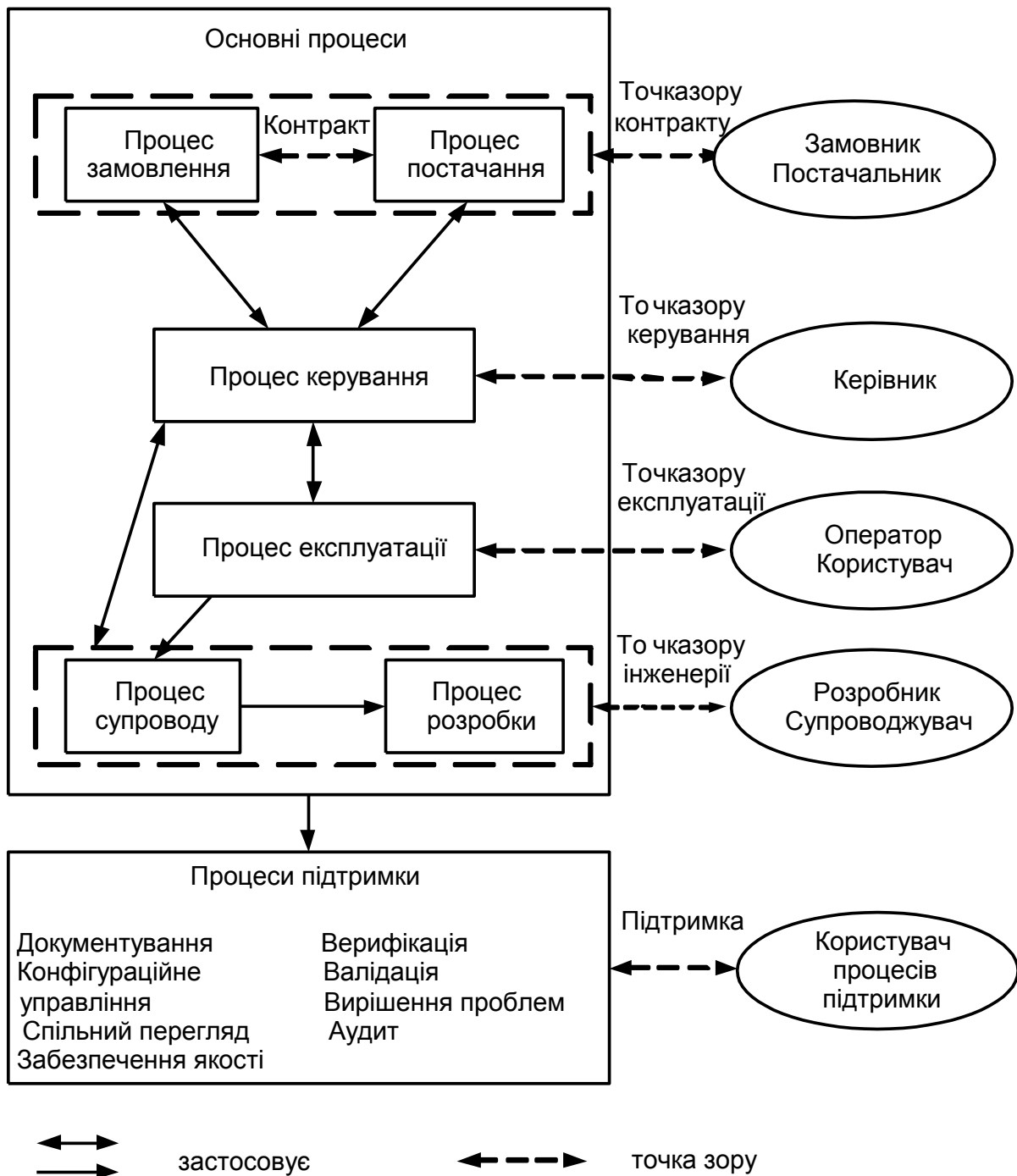


Рис. 12. Взаємозв'язки між процесами життєвого циклу

До **організаційних процесів** життєвого циклу відносяться:

керування – визначає основні дії щодо управління, включно з керуванням проектом, протягом життєвого циклу;

створення інфраструктури – визначає основні дії щодо створення основної структури процесів життєвого циклу;

удосконалення – визначає базові дії, які організація (якою може бути замовник, постачальник, розробник, оператор, супроводжувач або

керівник іншого процесу) виконує з метою створення, вимірювання, контролю та вдосконалення процесів життєвого циклу, які вона проводить;

навчання – визначає дії щодо забезпечення відповідного навчання персоналу.

Процеси життєвого циклу програмного забезпечення і взаємозв'язку між ними можуть розглядатися з різних точок зору (див. рис. 12). Ці процеси розглядаються з позицій контракту, керування, експлуатації, інженерії та підтримки:

з позицій **контракту** два учасники – замовник і постачальник – вступають у переговори, укладають контракт, використовуючи при цьому, відповідно, процес замовлення і процес постачання;

з позицій **управління** замовник, постачальник, розробник, оператор, супровідник та інші учасники управляють своїми процесами;

з позицій **експлуатації** оператор забезпечує службу експлуатації програмного забезпечення для користувачів;

з позицій **інженерії** розробник або супровідник виконують свої інженерні завдання щодо створення й модифікації програмного продукту;

з позиції **підтримки** учасники, які забезпечують, наприклад конфігураційне керування або забезпечення якості і т. д., забезпечують службу підтримки для інших учасників під час виконання ними специфічних, окремих завдань.

Процеси життєвого циклу – основні, підтримки, організаційні – включають певні дії. Дії основних процесів з різних точок зору наведені на рис. 13:

з погляду **контракту** використовуються два процеси життєвого циклу: процес замовлення – замовником, і процес постачання – постачальником. Ці процеси визначають завдання, відповідно, замовника і постачальника з погляду контракту;

з **інженерної** точки зору використовуються два процеси життєвого циклу – процес розробки і супроводу.

1. Основні процеси життєвого циклу

ТОЧКА ЗОРУ КОНТРАКТУ

1.1. Процес замовлення

Ініціювання

Підготовка запити щодо пропозицій

Підготовка та поновлення контракту

Нагляд за постачальником

Приймання та завершення

1.2. Процес постачання

Ініціювання

Підготовка відповіді

Укладення контракту

Планування

Виконання та контроль

Перегляд та оцінювання

Постачання та завершення

ІНЖЕНЕРНА ТОЧКА ЗОРУ

1.3. Процес розробки

Реалізація процесу

Інсталяція ПЗ

Забезпечення приймання ПЗ

Аналіз системних вимог

Проектування архітектури системи

Системна інтеграція

Системні кваліфікаційні випробування

Аналіз вимог до ПЗ

Проектування архітектури ПЗ

Деталізоване проектування ПЗ

Інтеграція ПЗ

Кваліфікаційні випробування ПЗ

Кодування та тестування ПЗ

1.4. Процес супроводу

Реалізація процесу

Аналіз проблем та модифікації

Реалізація модифікації

Перегляд-приймання модифікації

Перенесення

Вилучення ПЗ

ТОЧКА ЗОРУ ЕКСПЛУАТАЦІЇ

1.5. Процес експлуатації

Реалізація процесу

Експлуатаційні випробування

Експлуатація системи

Кодування та тестування ПЗ

Рис. 13. Дії основних процесів життєвого циклу ПЗ з різних точок зору

На рис. 14 наведені дії процесів підтримки життєвого циклу. З погляду **управління якістю** використовуються п'ять процесів життєвого циклу: процес забезпечення якості, процес верифікації, процес валідації, процес спільних розглядів і процес аудиту.

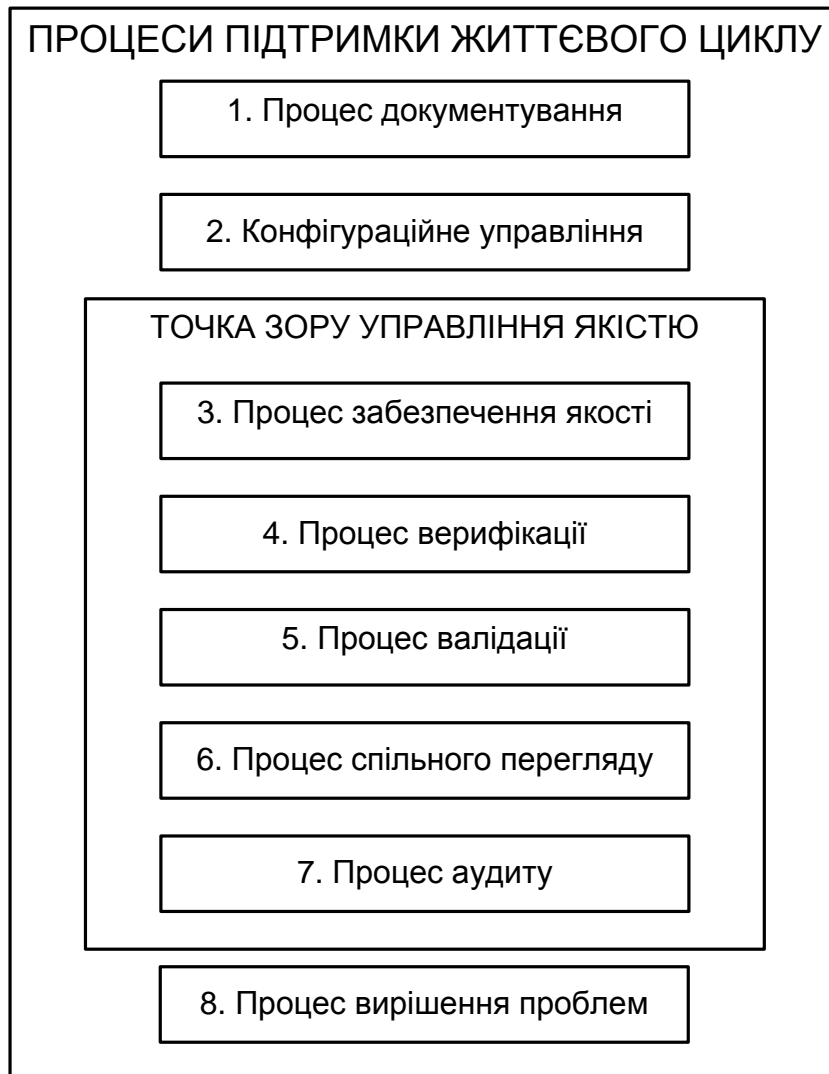


Рис. 14. Дії процесів підтримки ЖЦ ПЗ

Процеси, пов'язані з якістю, використовуються для управління якістю впродовж всього життєвого циклу програмного забезпечення. Процеси верифікації, валідації, спільних розглядів і аудиту можуть використовуватися різними учасниками окремо або як метод у рамках процесу забезпечення якості.

Слід більш детально розглянути процеси верифікації та валідації, враховуючи їх важливість у життєвому циклі ПЗ.

Процес верифікації – це процес визначення задоволення програмними продуктами вимог та умов, висунутих до них попередніми діями. З метою досягнення найбільшої ефективності з точки зору витрат та продуктивності верифікацію слід об'єднувати з процесами, які їх застосовують, такими як постачання, розробка, експлуатація, супровід.

Цей процес може виконуватися з різним ступенем незалежності. Ступінь незалежності може змінюватись залежно від того, виконується процес верифікації однією особою чи різними особами з однієї організації, чи він виконується особами з різних організацій, що різною мірою відокремлені одна від одної. У разі якщо процес виконується організацією незалежною від постачальника, розробника, оператора або супроводжувача, він називається **процесом незалежної верифікації**.

Процес верифікації складається з таких дій:

реалізації процесу;

верифікації.

Дія верифікації полягає у виконанні таких завдань.

Верифікація контракту здійснюється на основі розгляду таких критеріїв: постачальник має можливість задовольнити вимоги контракту; вимоги є несуперечливими та покривають потреби користувачів і т. ін.

Верифікація процесу здійснюється на основі розгляду таких критеріїв: вимоги щодо планування проекту є адекватними та вчасними; вибрані для проекту процеси є адекватними реалізованим і виконуються згідно з планом та відповідають контракту; стандарти, процедури та середовище проекту є адекватними (проект забезпечений персоналом і персонал проходить навчання згідно з контрактом).

Верифікація вимог здійснюється на основі розгляду таких критеріїв: системні вимоги несуперечливі, здійсненні та тестопридатні; системні вимоги відповідним чином розподілені між елементами апаратного забезпечення, елементами програмного забезпечення та ручними операціями згідно з проектними критеріями; вимоги до програмного забезпечення несуперечливі, здійсненні, тестопридатні та точно відображають системні вимоги; вимоги до програмного забезпечення, пов'язані з безпекою, захистом та критичністю, є коректними, що демонструється за допомогою відповідних точних методів.

Верифікація проекту здійснюється на основі розгляду таких критеріїв: проект коректний, а також несуперечливий і простежуваний

щодо вимог; проект реалізує правильну послідовність подій, входів-виходів, логіку функціонування, розподіл часових та обсягових ресурсів, а також визначення, локалізацію та відновлення системи після помилок; вибраний проект може бути виведений із вимог; проект коректно реалізує вимоги щодо безпеки, захисту та інші критичні вимоги, це демонструється за допомогою відповідних точних методів.

Верифікація коду здійснюється на основі розгляду таких критеріїв: код є простежуваним щодо проекту та вимог, тестопридатним, коректним і відповідає вимогам та стандартам кодування; код реалізує правильну послідовність подій, несуперечливі інтерфейси, коректні потоки даних та керування, повноту, відповідний розподіл часових та об'ємних ресурсів, а також визначення, локалізацію та відновлення системи після помилок.

Процес валідації – це процес визначення відповідності вимог та кінцевої побудованої системи (програмного продукту) встановленому для них призначенню. Цей процес може проводитись як складова частина дій щодо забезпечення приймання системи і складається з дій – реалізації процесу й валідації.

Дія валідації полягає у виконанні таких завдань:

підготовка вибраних вимог до тестів, тестових наборів даних та специфікацій тестів для аналізу результатів тестування;

забезпечення відображення детальних вимог до певного використання за призначенням у тестах, тестових наборах даних та специфікаціях тестів;

проведення тестування: на стресових, граничних та вироджених вхідних даних; програмного продукту на його здатність локалізувати та мінімізувати вплив помилок, тобто стійкість до відмов; запит на втручання оператора в стресових, граничних або вироджених умовах; на успішність вирішення окремими користувачами поставлених завдань;

підтвердження того, що програмний продукт відповідає своєму використанню за призначенням.

тестування програмного продукту як прийняттого в межах обраної сфери цільового середовища.

На рис. 15 наведені організаційні процеси життєвого циклу. З погляду **управління** використовується один процес – процес управління, який використовується будь-якою організацією для управління своїми процесами.



Рис. 15. Дії організаційних процесів ЖЦ ПЗ

Для підтримки практичного застосування стандарту ISO/IEC 12207 розроблений ряд технологічних документів: "Керівництво для ISO/IEC 12207 (ISO/IEC TR 15271:1998 Information technology – Guide for ISO/IEC 12207)" і "Керівництво щодо застосування ISO/IEC 12207 до управління проектами (ISO/IEC TR 16326:1999 Software engineering – Guide for the application of ISO/IEC 12207 to project management)".

Динамічність ІТ-галузі вимагала подальшого опрацювання стандарту ISO 12207, тому в 1998 р. вийшов новий стандарт ISO/IEC TR 15504: Information Technology – Software Process Assessment – Оцінка процесів розробки ПЗ. У цьому документі розглядаються питання атестації, визначення зрілості й удосконалення процесів життєвого циклу ПЗ. Один із розділів документа містить нову класифікацію процесів життєвого циклу, що є розвитком стандарту ISO 12207.

Зв'язок зі стандартом ISO 12207 полягає в тому, що всі процеси стандарту ISO 15504 належать до одного з наступних типів (табл. 4):

- базовий процес – процес зі стандарту 12207;
- розширений процес – розширення процесу зі стандарту 12207;
- новий процес – процес, не описаний у стандарті 12207;
- складовий процес – частина процесу зі стандарту 12207;

розширений складовий процес – розширена частина процесу зі стандарту 12207.

Таблиця 4

Процеси життєвого циклу ПЗ і систем за ISO 15504

Процеси "замовник – постачальник"	Процеси інженерні	Процеси допоміжні	Процеси управлінські	Процеси організаційні
придбання ПЗ; укладання контракту; визначення потреб замовника; проведення спільних експертиз і аудитів; підготовка до передачі; постачання і розгортання; підтримка експлуатації; надання послуг; оцінка вдовolenості замовників	виділення системних вимог і проектування системи загалом; виділення вимог до ПЗ; проектування ПЗ; реалізація, інтеграція і тестування ПЗ; інтеграція і тестування системи; супровід системи і ПЗ	розробка документації; управління конфігурацією; Забезпечення якості; вирішення проблем; проведення експертиз	планування життєвого циклу; планування проекту; побудова команди; управління вимогами; управління якістю; управління ризиками; управління ресурсами і графіком робіт; управління підрядниками	розвиток бізнесу; визначення процесів; вдосконалення процесів; навчання; забезпечення перевикористання; забезпечення інструментами; забезпечення середовища для роботи

У стандарті ISO 15504 запроваджується нова класифікація процесів:

1) основні процеси:

а) CUS: замовник-постачальник;

б) ENG: інженерні;

2) допоміжні процеси:

а) SUP: допоміжні;

3) організаційні процеси:

а) MAN: управлінські;

б) ORG: організаційні.

До процесів **замовник-постачальник** належать процеси, що безпосередньо впливають на замовника, підтримують процес розробки програмного забезпечення та його передачі споживачеві і що забезпечують можливість коректного використання програмного забезпечення.

Інженерні процеси складаються з процесів, які безпосередньо визначають, реалізують або підтримують програмний продукт, його

взаємодію з системою і документацію на нього. У тих випадках, коли система цілком складається з програмних засобів, інженерні процеси мають відношення лише до створення і підтримки цих програмних засобів.

Допоміжні процеси включають процеси, якими можуть користуватися будь-які інші процеси (включаючи інші допоміжні процеси) в різні моменти життєвого циклу програмного забезпечення.

Управлінські процеси включають процеси, що містять практики загального характеру, які можуть бути використані кожним, хто управляє будь-яким проектом або процесом у ході життєвого циклу програмного забезпечення.

До **організаційних процесів** відносяться процеси, які встановлюють цілі функціонування організації і створюють активи процесів, продуктів та ресурсів, які, використовуючись у проектах організації, сприяють виконанню її цілей. Хоча організаційні процеси загалом відносяться не тільки до програмних засобів, вони виконуються в загальному контексті організації і для їх ефективного використання необхідне відповідне оточення. Організаційні процеси використовують усе краще з того, що є в організації: передовий досвід, ефективні процеси, кращі навички, якісний програмний код, хороші засоби підтримки процесів для створення інфраструктури організації.

Стандарти організацій ISO і IEEE побудовані на основі наявного емпіричного досвіду розробки, отриманого в рамках поширених якийсь час тому концепцій та інструментальних засобів. Це не означає, що вони застаріли, оскільки їх автори мають достатньо добре уявлення про нові методи й технології розробки і намагаються дивитися вперед. Але в процесі використання новаторських технологій у розробці ПЗ може так статися, що частина вимог стандарту забезпечується абсолютно іншими способами, ніж це передбачено в ньому, а частина артефактів взагалі не присутня в рамках даної технології, зникнувши внаслідок автоматизації процесів.

9.2.5. Стандарти зрілості можливостей організації, розроблені SEI

Модель CMM. Стандарт визначає, яким ознакам повинна відповідати професійна організація з розробки ПЗ. Модель CMM пропонує уніфікований підхід до оцінки можливостей організації

виконувати завдання різного рівня. Для цього визначаються три рівні елементів:

- рівні зрілості організації (maturity levels);
- ключові області процесу (key process areas);
- ключові практики (key practices).

Найчастіше під моделлю CMM мають на увазі саме модель рівнів зрілості. Професіоналізм організації визначається за допомогою зрілості процесу, використовуваного цією організацією. У стандарті виділяються п'ять рівнів зрілості процесу, які встановлюють ступінь готовності організації виконати великий проект.

Рівні зрілості. CMM описує різні ступені зрілості процесів в організаціях, визначаючи п'ять рівнів організацій:

початковий (initial) – технологія повністю імпровізована, іноді навіть хаотична; успіх цілком залежить від зусиль окремих співробітників. До цього рівня належать організації, що розробляють ПЗ, але які не мають усвідомленого процесу розробки, не проводять планування й оцінку своїх можливостей;

повторюваний (repeatable) – базові процеси управління проектом ПЗ установлені; є дисципліна дотримання, що забезпечує можливість повторення успіху попередніх проектів у тій же прикладній області. В організаціях, що відповідають цьому рівню, ведеться облік витрат ресурсів і відстежується хід проектів, встановлені правила управління проектами, засновані на отриманому досвіді;

визначений (defined) – процеси задокументовані, стандартизовані й інтегровані в єдину для всієї організації технологію створення ПЗ; для кожного проекту використовується адаптивний варіант цієї технології. В організаціях цього рівня є прийнятий, повністю документований, відповідний реальному стану справ і доступний персоналу процес розробки й супроводу ПЗ. Він повинен включати як управлінські, так і технічні підпроцеси, а також навчання співробітників роботі в його рамках;

керований (manageable) – збираються й накопичуються метрики (об'єктивні дані) про якість виконання процесів і вихідної продукції; управління процесами і вихідною продукцією здійснюється за кількісними оцінками. В організаціях цього рівня, крім встановленого і описаного процесу, використовуються вимірні показники якості продуктів і результативності процесів, що дозволяють достатньо точно передбачати

обсяг ресурсів (часу, грошей, персоналу), необхідний для розробки продукту з певною якістю;

що вдосконалюється (optimizing) – вдосконалення технології створення ПЗ здійснюється безперервно на основі кількісного зворотного зв'язку від процесів і пілотного впровадження інноваційних ідей. У таких організаціях, крім процесів і методів їх оцінки, є методи визначення слабких місць, визначені процедури пошуку й оцінки нових методів і техніки розробки, навчання персоналу роботі з ними і їх включення до загального процесу організації в разі підвищення ними ефективності виробництва.

Ключові області процесу. Згідно із СММ, рівні зрілості організації можна визначати за використанням в організації чітко встановленої техніки і процедур, що відносяться до різних ключових галузей процесу. Кожна така область є набором пов'язаних видів діяльності, націлених на досягнення цілей, суттєвих для оцінки результативності технологічного процесу в цілому. Всього виділяється 18 областей. Множина ділянок, які повинні підтримуватися організацією, розширюється в процесі переходу до вищих рівнів зрілості.

До першого рівня не висуваються жодні вимоги.

Організації другого рівня зрілості повинні певним чином підтримувати управління вимогами, планування проектів, нагляд за ходом проекту, управління підрядниками, забезпечення якості ПЗ, управління конфігурацією.

Організації третього рівня повинні, крім діяльностей другого рівня, підтримувати проведення експертиз, координацію діяльності окремих груп, розробку програмного продукту, інтегроване управління розробкою і супроводом, навчання персоналу, вироблення і підтримку технологічного процесу організації, контроль за дотриманням технологічного процесу організації.

До діяльностей організацій четвертого рівня додаються управління якістю ПО та управління процесом, засноване на вимірних показниках.

Організації п'ятого рівня зрілості повинні додатково підтримувати управління змінами процесу, управління змінами використовуваних технологій і запобігання дефектам.

Ключові практики. Ключові області процесу описуються за допомогою наборів ключових практик. Ключові практики розподіляються на декілька видів:

зобов'язання (commitments to perform);

можливості (abilities to perform);

діяльності (activities performed);
вимірювання (measurements and analysis);
перевірки (verifying implementations).

Наприклад, управління вимогами пов'язане з наступними практиками:

зобов'язання – проекти повинні відповідати певній політиці організації щодо управління вимогами;

можливості – у кожному проекті повинен визначатися відповідальний за аналіз системних вимог і прив'язування їх до апаратного, програмного забезпечення та інших компонентів системи. Вимоги повинні бути документовані. Для управління вимогами повинні бути виділені адекватні ресурси і бюджет. Персонал повинен проходити навчання в області управління вимогами;

діяльності – перш ніж бути включеними в проект, вимоги підлягають аналізу на повноту, адекватність, несуперечність та ін. Виділені вимоги використовуються як основа для планування та виконання інших робіт. Зміни у вимогах аналізуються і включаються в проект;

вимірювання – проводиться періодичне визначення статусу вимог і статусу діяльності щодо управління ними;

перевірки – діяльність з управління вимогами періодично аналізується старшими менеджерами, діяльність з управління вимогами періодично і на підставі значущих подій аналізується менеджером проекту, група забезпечення якості проводить аналіз і аудит діяльності з управління вимогами і звітує за результатами цього аналізу.

Модель CMMI. На зміну моделі CMM, що зараз вважається застарілою, була розроблена модель CMMI. Ця модель є результатом інтеграції моделей CMM для продуктів і процесів, а також для розробки ПЗ і розробки програмно-апаратних систем. Основні зміни порівняно із CMM наступні. Зроблені два виклади моделі – безперервне й поетапне. Перший призначений для полегшення міграції від підтримки американського галузевого стандарту EIA/AIS 713 і поступового вдосконалення процесів за рахунок впровадження різних практик. Другий призначений для полегшення міграції від підтримки CMM і порівневого розгляду практик, що вводяться.

Елементи моделі розподіляють на три види:

обов'язкові (required);
такі, що рекомендуються (expected);
інформативні (informative).

Тип елемента чітко позначається в моделі.

Використовувані практики розподіляються на:
загальні (generic);

специфічні (specific).

Вони доповнюються набором загальних і специфічних цілей, яких необхідно досягти для забезпечення певного рівня зрілості в певних областях процесу.

Деякі рівні зрілості отримали інші назви. Другий рівень був названий **керованим (managed)**, а четвертий – **керованим на основі метрик (quantitatively managed)**.

Набір областей процесу і практик, що виділяються, значно змінився. Усі області процесу діляться на чотири категорії (області процесу, що наводиться нижче, помічені номером рівні зрілості, починаючи з якого вони повинні підтримуватися згідно із CMMI):

управління процесом, яке включає вироблення й підтримку процесу (3), контроль за дотриманням процесу (3), навчання (3), вимірювання показників процесу (4), впровадження інновацій (5);

управління проектом, яке включає планування проектів (2), контроль за ходом проекту (2), управління угодами з постачальниками (2), інтегроване управління проектами (3), управління ризиками (3), побудову команд (3), управління постачальниками (3) і вимірювання показників результативності і ходу проекту (4);

технічні, які включають формування вимог (3), управління вимогами (2), вироблення технічних рішень (3), інтеграцію продуктів (3), верифікацію (3) і валідацію (3);

підтримки включають управління конфігурацією (2), забезпечення якості продуктів і процесів (2), проведення вимірювань і аналіз їх результатів (2), управління оточенням (3), аналіз і ухвалення рішень (3), аналіз, вирішення і запобігання проблемам (5).

Усі розглянуті стандарти визначають певний набір видів діяльності, з яких повинен складатися процес розробки, і вводять ту чи іншу структуру на цих видах діяльності, виділяючи їх елементи. Разом з тим вони не можуть бути зведені без суттєвих змін в єдину модель життєвого циклу ПЗ. У цілому наявні стандарти слабо узгоджені між собою.

Крім того, дані стандарти не пропонують чітких і однозначних схем побудови життєвого циклу ПЗ, зокрема, зв'язків між окремими діяльностями. Це зроблено навмисно, оскільки раніше діючі стандарти були досить жорстко прив'язані до каскадної моделі життєвого циклу і таким чином перешкоджали використанню прогресивніших технологій

розробки. Сучасні стандарти прагнуть максимально загальним чином визначити набір видів діяльності, які повинні бути представлені в рамках життєвого циклу (з урахуванням цілей окремих проектів – тобто проект, в якому не прагнуть досягти якихось цілей, може не включати діяльностей, пов'язаних із їх досягненням), і описати їх за допомогою наборів вхідних документів і результатів.

Слід зауважити, що стандарти можуть мати досить великі розбіжності з реальними розробками, якщо в них використовуються новітні методи, інструментальні засоби та інформаційні технології розробки й супроводу ПЗ.

9.3. Традиційні моделі життєвого циклу програмного забезпечення

9.3.1. *Поняття моделі життєвого циклу програмного забезпечення*

Модель життєвого циклу програмного забезпечення (Software Life Cycle Model, SLCM) – це концептуальна структура, що включає процеси, дії та завдання, які стосуються розробки, експлуатації та супроводу програмного продукту, й охоплює життєвий цикл системи, починаючи з визначення вимог до неї й закінчуючи припиненням її використання. SLCM визначає точні інструкції, які розробник може використовувати для створення високоякісних програмних систем. Конкретні моделі життєвого циклу програмного забезпечення визначаються особливістю завдань, обмеженнями на ресурси, досвідом розробників і т. д.

Модель життєвого циклу ПЗ описується набором фаз (етапів, стадій) проекту зі створення ПЗ, в яких виконуються окремі процеси, розбиті на дії та завдання.

Фаза проекту – об'єднання логічно пов'язаних процесів проекту, що звичайно завершуються досягненням одного з основних результатів. Фаза може розбиватися на етапи. Відмінною особливістю фази проекту від процесу є те, що *фаза завершується отриманням одного з основних результатів, тоді як процес – просто значущого результату*. Слід зазначити, що склад, кількість і порядок виконання фаз визначається особливістю проекту.

Серед відомих моделей життєвого циклу можна виділити наступні:
каскадна модель;
V-подібна модель;
модель прототипування;
модель RAD;
інкрементна модель;
спіральна модель.
Вибір конкретної моделі ЖЦ ПЗ відбувається за схемою (рис. 16).

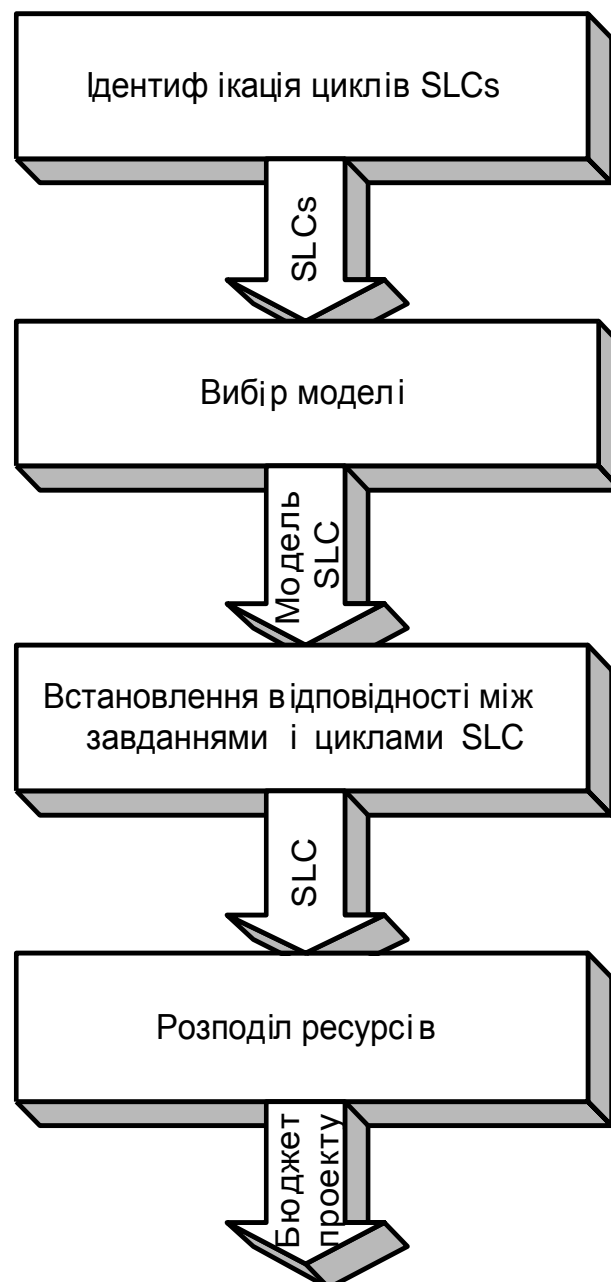


Рис. 16. Схема вибору моделі життєвого циклу ПЗ

9.3.2. Каскадна модель

Каскадна модель вперше чітко сформульована в 1970 році Ройсом. Спочатку ця модель мала значний вплив на розробку складного ПЗ, як метод, який регулює послідовність і зміст фаз розробки проекту. Це була перша модель, у якій формалізувалася структура етапів розробки ПЗ і надавалося велике значення вимогам, що висувалися до програмного продукту і процесів проектування, а також до документування на всіх етапах розробки проекту. У 70 – 80-х роках минулого століття ця модель була прийнята як стандарт Міністерства оборони США.

Каскадна модель (від англ. *waterfall* – водоспад) відповідно до стандарту IEEE 1074 включає виконання наступних фаз (рис. 17):

дослідження концепції, яке включає дослідження вимог, розробку бачення продукту і оцінювання можливості його реалізації;

дослідження системи, що виконується для систем, в яких необхідна розробка не тільки програмного, але й апаратного забезпечення. Функції, що розробляються, розподіляються для ПЗ і устаткування відповідно до архітектури системи;

визначення вимог, у ході якого специфікуються програмні вимоги для предметної області системи, визначаються цілі, функції, інтерфейси і продуктивність продукту.

розробки проекту, в процесі якої виконується логічна розробка технічних характеристик програмної системи, включаючи структури даних, архітектуру ПЗ, інтерфейсні подання і процесну (алгоритмічну) деталізацію;

реалізації, під час якої логічний опис ПЗ, виконаний у попередній фазі, перетворюється на повноцінний програмний продукт. Результат: початковий код, база даних і документація. У реалізації звичайно виділяють два етапи: реалізацію компонент ПЗ та інтеграцію компонент у готовий продукт. На обох етапах виконується кодування і тестування, які також інколи розглядають як два підетапи;

встановлення, яке включає встановлення ПЗ, його перевірку і офіційне приймання замовником;

експлуатації та підтримки, які мають на увазі запуск і поточне забезпечення, включаючи надання технічної допомоги, обговорення виниклих питань з користувачем, реєстрацію запитів користувача на

модернізацію і внесення змін, а також коректування або усунення помилок;

супроводу, тобто усунення програмних помилок, несправностей, збоїв, модернізація і внесення змін. Складається з ітерацій розробки;

виведення з експлуатації, тобто виведення системи з її активного використання шляхом припинення роботи або її заміни новою чи модернізованою системою.

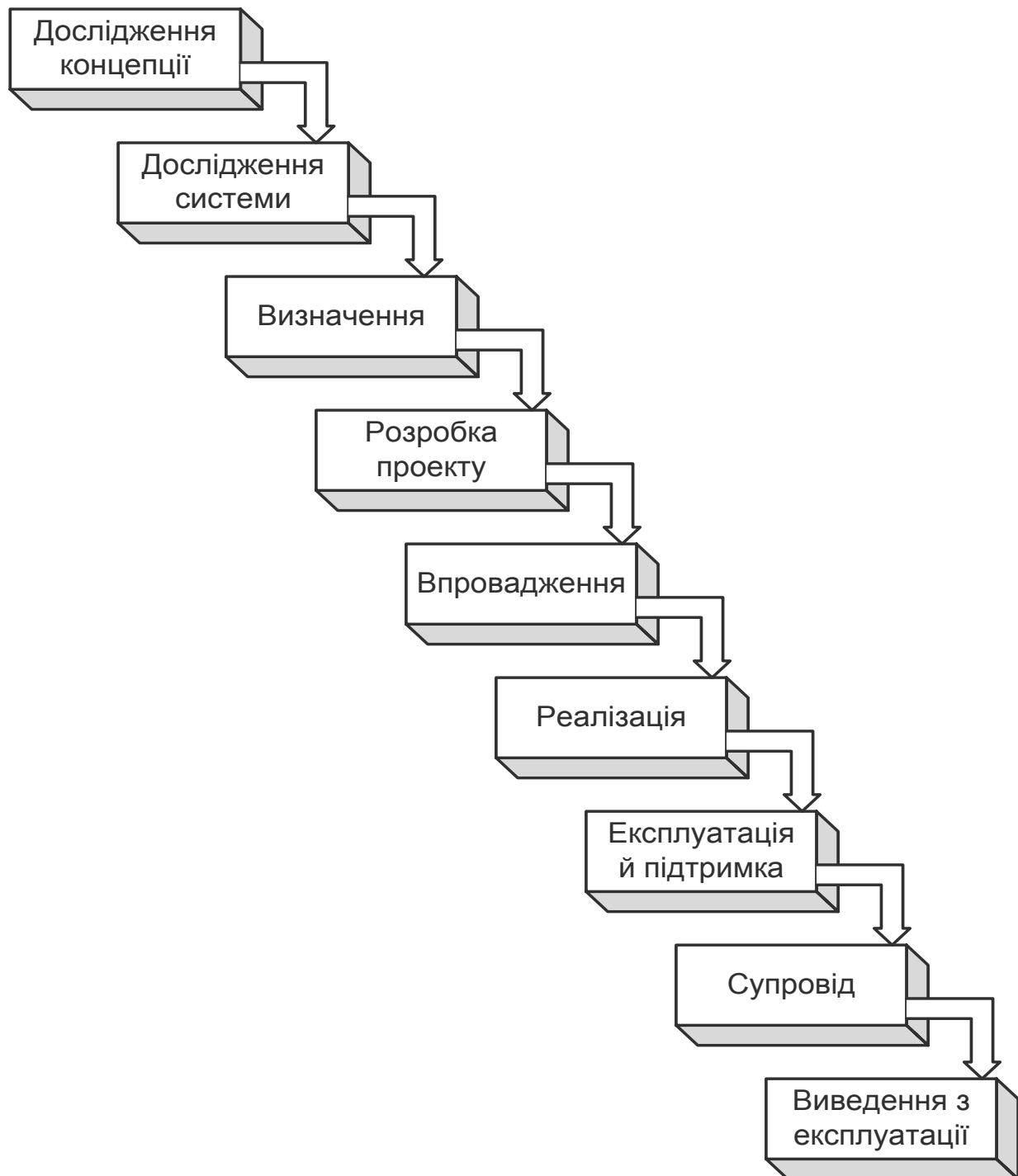


Рис. 17. Каскадна модель життєвого циклу ПЗ

Проектування на основі каскадної моделі повинне керуватися наступними **принципами**:

послідовне виконання фаз, тобто кожна наступна фаза починається, коли цілком завершено виконання попередньої фази;

кожна фаза має певні вхідні й вихідні дані;

кожна фаза повністю документується;

перехід від однієї фази до іншої здійснюється за допомогою формального огляду за участю замовника;

основу моделі складають вимоги, сформульовані в документі SRS – "Специфікація вимог до програмного забезпечення" (або ТЗ – "Технічному завданні"), які змінюватися не повинні;

критерій якості проекту – відповідність створеного продукту вимогам, встановленим у SRS або ТЗ.

Каскадна модель має наступні **переваги**:

проста й зрозуміла замовникам, оскільки часто використовується організаціями для відстеження проектів, не пов'язаних із розробкою ПЗ;

проста й зручна в застосуванні, тому що процес розробки виконується поетапно;

її структурою може керуватися навіть слабо підготовлений або недосвідчений в технічному плані персонал;

вона сприяє здійсненню чіткого контролю за управлінням проектом;

кожну стадію можуть виконувати незалежні команди, тому що процес розробки обов'язково документується;

полегшує планування термінів і витрат на розробку.

Для каскадної моделі життєвого циклу характерна автоматизація окремих незв'язаних завдань, які не потребують виконання інформаційної інтеграції та сумісності, програмного, технічного й організаційного поєднання. У рамках вирішення окремих завдань каскадна модель життєвого циклу за термінами розробки й надійністю виправдала себе. Застосування каскадної моделі життєвого циклу до великих і складних проектів унаслідок великої тривалості процесу проектування і мінливості вимог за цей час приводить до неможливості їх практичної реалізації.

Використання каскадної моделі для проектів, у яких вимоги і їх реалізація не можуть бути достатньо чітко визначені або можуть змінюватися в процесі виконання проекту, стикається з наступними основними її недоліками:

спроба повернутися на одну або дві фази назад, щоб виправити яку-небудь проблему або недолік, призведе до значного збільшення витрат і збою в графіку;

інтеграція компонент, у процесі якої звичайно виявляється велика частина помилок, виконується в кінці розробки, що сильно збільшує вартість усунення помилок;

запізнювання з отриманням результатів – якщо в процесі виконання проекту вимоги змінилися, то результат буде застарілим.

Каскадна модель з можливістю повернення на попередній крок за необхідності переглянути його результати стає **ітераційною** (рис. 18).

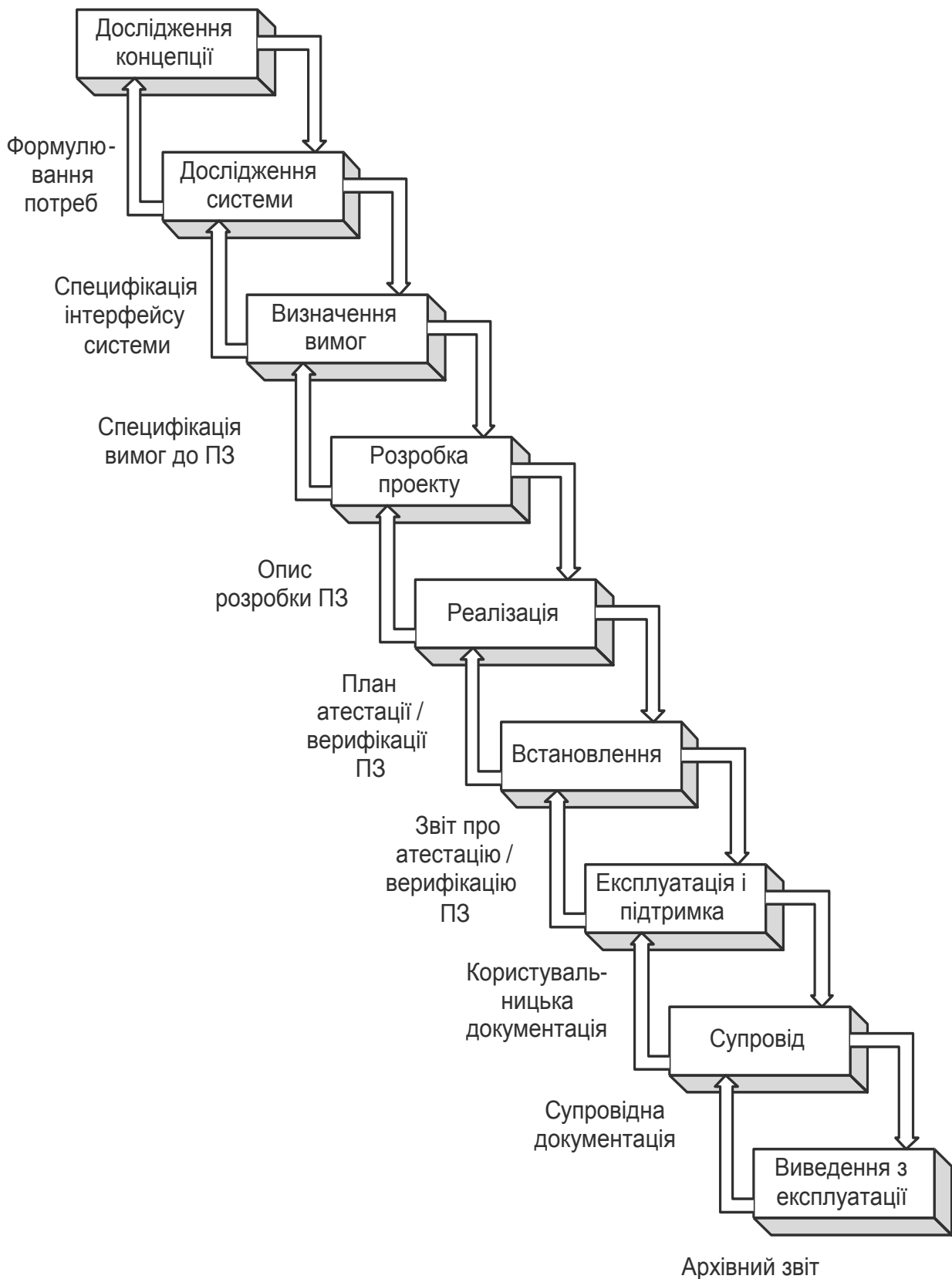


Рис. 18. Каскадна модель життєвого циклу ПЗ зі зворотними зв'язками

Недоліки каскадної моделі привели до того, що на сьогодні вона мало застосовується. Тому застосування каскадної моделі обмежується вирішенням завдань, коли вимоги і їх реалізація максимально чітко визначені та зрозумілі. До них відносяться завдання наступних типів:

науково-обчислювального характеру (пакети й бібліотеки наукових програм розрахункового характеру в будівництві, автобудуванні і т. д.);

проекування операційних систем і компіляторів;

систем реального часу управління конкретними об'єктами.

Слід особливо зазначити, що принципи каскадної моделі є основою для елементів моделей інших типів.

9.3.3. V-подібна модель

V-подібна модель була розроблена як різновид каскадної моделі. У ній більше уваги приділяється плануванню проекту з можливістю паралельного тестування системи. Особливе значення надається верифікації та атестації продукту, тому що тестування продукту обговорюється, проектується і планується на ранніх етапах життєвого циклу розробки. План випробування приймання замовником розробляється на етапі планування, а компонування випробування системи – на фазах аналізу, розробки проекту і т. д. Цей процес розробки планів випробування позначений пунктирною лінією між прямокутниками V-подібної моделі (рис. 19).

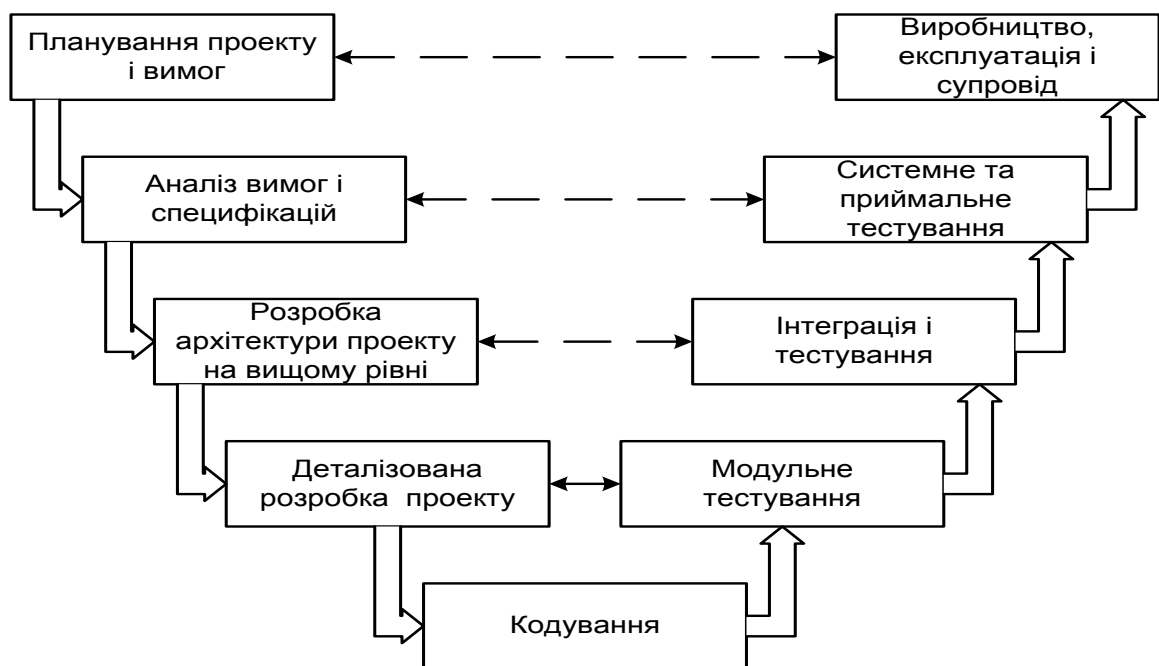


Рис. 19. V-подібна модель життєвого циклу

V-подібна модель успадкувала від каскадної таку ж послідовну структуру. Кожна наступна фаза починається після завершення отримання результативних даних попередньої фази. Модель демонструє комплексний підхід до визначення фаз процесу розробки ПЗ. У ній підкреслені взаємозв'язки, що існують між аналітичними фазами і фазами проектування, які передують кодуванню, після якого йдуть фази тестування. Пунктирні лінії між фазами означають, що їх необхідно розглядати паралельно.

У V-подібній моделі передбачене виконання наступних фаз:

планування проекту і вимог, коли визначаються системні вимоги, розподіляються ресурси організації з метою їх відповідності поставленим вимогам;

аналізу вимог до продукту і його специфікації, тобто аналіз вимог до ПЗ завершується цілковитою їх специфікацією;

високорівневої розробки архітектури, яка визначає функції ПЗ, які будуть реалізовані в проекті;

деталізованої розробки проекту, яка визначає алгоритми для кожного компоненту, який був визначений на фазі побудови архітектури. Ці алгоритми надалі будуть перетворені в код;

розробки програмного коду, в процесі якої виконується перетворення алгоритмів, визначених на етапі деталізованого проектування, в готове ПЗ;

модульного тестування, у ході якого виконується перевірка кожного закодованого модуля на наявність помилок;

інтеграції та тестування, під час якого виконується перевірка на наявність помилок в інтегрованій групі модулів;

системного й приймального тестування, в ході якого виконується перевірка функціонування програмної системи в цілому (повністю інтегрованої системи) після розміщення її в апаратному середовищі відповідно до специфікації вимог до ПЗ;

виробництва, приймальних випробувань, експлуатації та супроводу, коли ПЗ запускається у виробництво, користувач тестує функціональні можливості системи на відповідність початковим вимогам. Після цього забезпечується супровід системи.

Переваги V-подібної моделі:

у моделі особливе значення надається плануванню, спрямованому на верифікацію й атестацію продукту, що розробляється, на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції і тестування реалізують архітектурне проектування або проектування на вищому

рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації; у моделі передбачені атестація і верифікація всіх зовнішніх та внутрішніх отриманих даних, а не лише самого програмного продукту;

визначення вимог виконується перед розробкою проекту системи, а проектування ПЗ – перед розробкою компонентів;

модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожні отримані дані повинні підлягати тестуванню;

менеджер проекту може відстежувати хід процесу розробки, оскільки в даному випадку цілком можливо скористатися часовою шкалою, а завершення кожної фази є контрольною точкою;

модель проста у використанні.

Недоліки V-подібної моделі:

труднощі під час роботи з паралельними подіями;

не враховані ітерації між фазами;

не передбачене внесення динамічних змін на різних етапах життєвого циклу;

тестування вимог в життєвому циклі відбувається надто пізно, внаслідок чого неможливо внести зміни, без зміни графіка виконання проекту;

у модель не входять дії, спрямовані на аналіз ризиків.

Сфера застосування V-подібної моделі. V-подібну модель краще всього використовувати, як і у випадку з каскадною моделлю, коли вся інформація про вимоги доступна заздалегідь. Використання моделі ефективне, коли доступними є інформація про метод реалізації розв'язання і технологія, а персонал володіє необхідними вміннями і досвідом роботи з даною технологією. V-подібну модель звичайно застосовують, коли потрібна висока надійність систем, пов'язана з ризиком для життя і т. д.

9.3.4. Модель швидкого прототипування

У 1975 р. Фред Брукс (Fred Brooks) у своїй книзі "Легендарна людина-місяць" ("The Mythical Man-Month") написав: "У більшості проектів перша побудована система навряд чи придатна до використання. Вона може бути занадто повільною, надто об'ємною,

незручною у використанні або мати всі три перераховані недоліки. Але немає іншого вибору, окрім як почати із самого початку, доклавши всіх зусиль, і побудувати модернізовану версію, в якій розв'язувалися б усі три проблеми...". Саме ця концепція побудови експериментальної, або прототипної системи привела до виникнення "структурної", "еволюційної" моделі швидкого прототипування (RAD) і спіральної моделі.

Прототипування – це процес побудови робочої моделі системи.

Прототип – еквівалент експериментальної моделі або "макету" в середовищі апаратного забезпечення. Виконання еволюційних програм відбувається в рамках контексту плану, спрямованого на досягнення гранично високої продуктивності. Цей метод передбачає також, що розробка інкрементів програми очевидна для користувача, який бере участь у всьому процесі розробки.

Модель швидкого прототипування (рис. 20) призначена для швидкого створення прототипів продукту з метою уточнення вимог і поетапного розвитку прототипів у кінцевий продукт. Швидкість (висока продуктивність) виконання проекту забезпечується плануванням розробки прототипів і участю замовника в процесі розробки.

Перший рівень. Початок життєвого циклу розробки поміщений у центрі еліпса. Спільно з користувачем розробляється попередній план проекту на основі попередніх вимог. Результат початкового планування – документ, що описує в загальних рисах приблизні графіки й результівні дані.

Другий рівень – створення початкового прототипу на основі швидкого аналізу, проектування бази даних, призначеного для користувача інтерфейсу і деяких функцій.

Третій рівень. Починається ітераційний цикл швидкого прототипування. Розробник проекту демонструє черговий прототип, користувач оцінює його функціонування, спільно визначаються проблеми і шляхи їх подолання для переходу до наступного прототипу. Цей процес триває доти, поки користувач не погодиться, що черговий прототип точно відображає всі вимоги.

Четвертий рівень. Отримавши схвалення користувача, швидкий прототип перетворюють у детальний проект, і систему налаштовують на виробниче використання. Саме на цьому етапі налаштування прискорений прототип стає системою, що повністю діє.

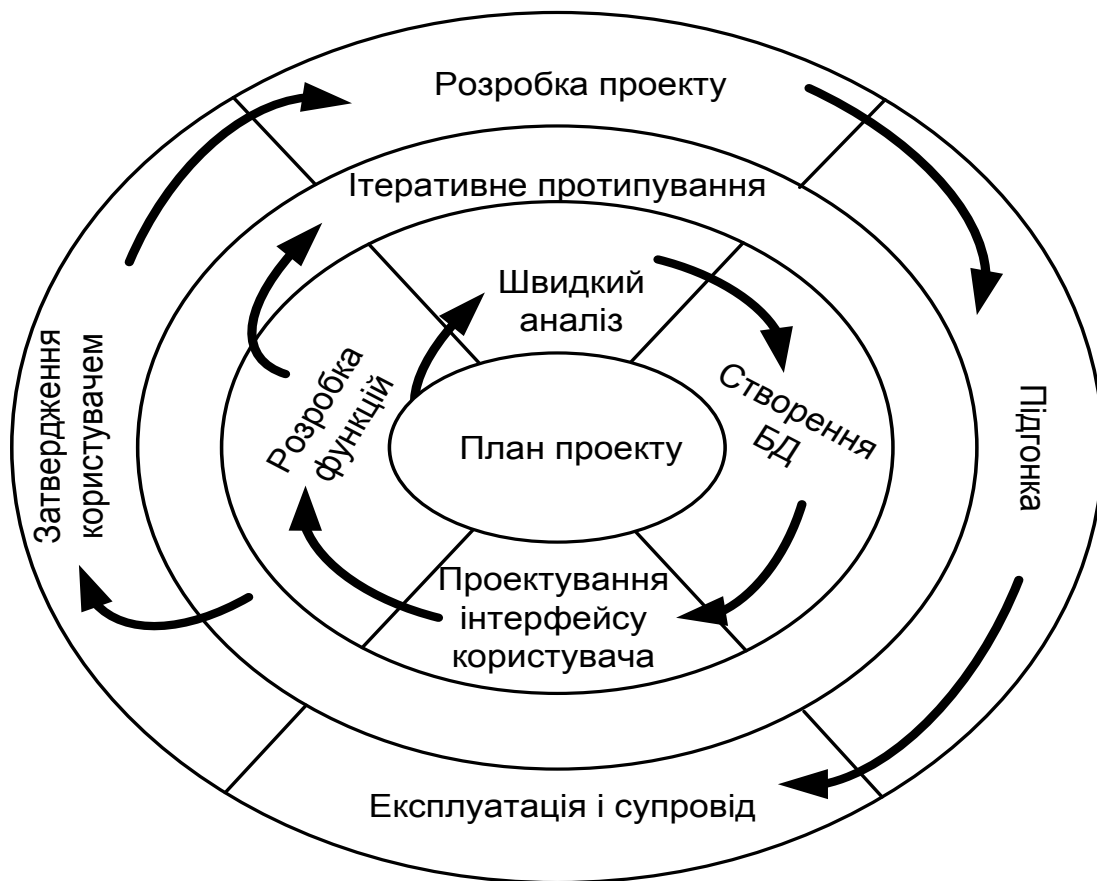


Рис. 20. **Модель швидкого прототипування**

Під час розробки виробничої версії програми може знадобитися вищий рівень функціональних можливостей, різні системні ресурси, необхідні для забезпечення повного робочого навантаження або обмеження в часі. Після цього йдуть тестування в граничних режимах, визначення вимірювальних критеріїв і налаштування, а потім, як завжди, функціональний супровід.

Переваги моделі швидкого прототипування:

кінцевий користувач може спостерігати системні вимоги в процесі їх збирання командою розробників; таким чином, взаємодія замовника із системою починається на ранньому етапі розробки;

виходячи з реакції замовників на демонстрації продукту, що розробляється, розробники отримують відомості про один або декілька аспектів поведінки системи, завдяки чому зводиться до мінімуму кількість неточностей у вимогах;

у процес розробки можна внести нові або неочікувані вимоги користувача;

можна виконувати гнучке проектування й розробку, включаючи декілька ітерацій на всіх фазах життєвого циклу;

при використанні моделі спостерігаються видимі ознаки прогресу у виконанні проекту, завдяки чому замовники відчують себе впевнено;
можливість виникнення розбіжностей у процесі спілкуванні замовників із розробниками мінімізована;
очікувана якість продукту визначається за активної участі користувача на ранніх фазах розробки;
можливість спостерігати ту чи іншу функцію в дії сприяє розробці додаткових функціональних можливостей;
завдяки меншому обсягу доробок зменшуються витрати на розробку;
завдяки тому, що проблема виявляється до залучення додаткових ресурсів, скорочуються загальні витрати;
забезпечується управління ризиками;
документація сконцентрована на кінцевому продукті, а не на його розробці.

Недоліки моделі швидкого прототипування:

швидко створені прототипи страждають від неадекватної документації або її браку;

якщо цілі прототипування не узгоджені заздалегідь, процес проектування може стати некерованим;

під час створення робочого прототипу якості всього ПЗ або довгостроковій експлуатаційній надійності може бути приділено недостатньо уваги;

іноді внаслідок використання моделі отримують систему з низькою робочою характеристикою, особливо якщо в процесі її виконання передбачається етап підгонки;

у процесі використання моделі вирішення складних проблем може відкладатися на майбутнє. У результаті це призводить до того, що наступні отримані продукти можуть не виправдати надій, які покладалися на прототип;

якщо користувачі не можуть брати участі в проекті на ітераційній фазі швидкого прототипування життєвого циклу, це може відбитися на якості кінцевого продукту;

на ітераційному етапі прототипування швидкий прототип є частковою системою. Якщо виконання проекту завершується достроково, у кінцевого користувача залишиться тільки часткова система;

замовник може вважати за краще отримати прототип, замість того, щоб чекати появи повної, добре продуманої версії;

і розробники, і користувачі не завжди розуміють, що коли прототип перетворюється на кінцевий продукт, необхідне створення традиційної документації. Якщо вона відсутня, модифікувати модель на пізніших етапах може виявитися дорожчим заняттям, ніж просто не скористатися створеним прототипом;

на розробку системи може бути витрачено надто багато часу, тому що ітераційний процес демонстрації прототипу і його перегляд можуть тривати нескінченно без належного управління процесом.

Сфера застосування структурної еволюційної моделі швидкого прототипування:

- вимоги невідомі заздалегідь;

- вимоги непостійні або можуть неправильно тлумачитися чи бути невдало сформульованими;

- слід уточнити вимоги;

- існує потреба в розробці призначених для користувача інтерфейсів;

- потрібна перевірка концепції;

- здійснюються тимчасові демонстрації;

- виконується нова, яка не має аналогів, розробка (на відміну від експлуатації продукту на вже існуючій системі);

- потрібно зменшити неточності у визначенні вимог, тобто зменшується ризик створення системи, яка не має жодної цінності для замовника;

- вимоги схильні до швидких змін, коли замовник неохоче погоджується на фіксований набір вимог або якщо про прикладну програму відсутнє чітке уявлення;

- розробники невпевнені в тому, яку оптимальну архітектуру або алгоритм слід застосовувати;

- алгоритми або системні інтерфейси ускладнені;

- потрібно продемонструвати технічну здійсненність, коли технічний ризик високий.

Швидке прототипування особливо добре підходить для розробки інтенсивно використовуваних систем призначеного для користувача інтерфейсу, таких як інтерактивні системи, нові у своєму роді продукти, а також системи забезпечення ухвалення рішень, наприклад, управління, медична діагностика.

9.3.5. Модель швидкої розробки ПЗ (RAD)

Метод швидкої розробки додатків (Rapid Application Development, RAD) був використаний у 80-х роках ХХ століття фірмою IBM. Цей метод був уперше запропонований до уваги розробників ПЗ у книзі Джеймса Мартіна. Завдяки методу RAD користувач задіяний на всіх фазах життєвого циклу розробки проекту не лише під час визначення вимог, але й у процесі проектування, розробки, тестування, а також кінцевого постачання програмного продукту. Участь користувача в процесі розробки стає такою активною завдяки використанню засобів розробки та середовища, що дозволяє дати оцінку продукту на всіх стадіях його розробки. Це забезпечується наявністю *засобів розробки графічного, призначеного для користувача інтерфейсу і кодогенераторів*.

Характерною рисою RAD є короткий час переходу від визначення вимог до створення цілісної системи. Метод ґрунтується на послідовній ітерації еволюційної системи або прототипів, критичний аналіз яких обговорюється із замовником. У процесі такого аналізу формуються вимоги до продукту. Розробка кожного інтегрованого продукту обмежується чітко окресленим періодом часу, який, як правило, становить 60 днів і називається **часовим блоком**.

Чинники, що дозволяють створити систему за 60 днів без шкоди для якості, включають:

- використання потужних інструментальних засобів розробки;
- високий рівень повторного використання елементів системи;
- осмислені й виділені ресурси.

Вирішальна рольова участь кінцевого користувача полягає в переміщенні процесу роботи від програмування і тестування до планування та проектування. Користувачам доводиться виконувати великий обсяг роботи на початку життєвого циклу, але за це вони отримують систему, побудовану за коротший проміжок часу.

Фази моделі ЖЦ RAD і участь користувача на всіх фазах процесу проектування наведені на рис. 21:

планування вимог – збір вимог виконується в процесі використання методу, який називають спільним плануванням вимог (Joint requirements planning, JRP), який є структурним аналізом і обговоренням наявних комерційних завдань;

призначений для користувача опис – спільне проектування додатка (Joint application design, JAD) використовується з метою залучення користувачів; на цій фазі проектування системи, яка не є промисловою, команда, яка працює над проектом, часто використовує

інструментальні засоби, що забезпечують збирання призначеної для користувача інформації;

конструювання – ця фаза об'єднує в собі деталізоване проектування, побудову (кодування і тестування), а також постачання програмного продукту замовникові за певний час. Терміни виконання цієї фази значною мірою залежать від використання генераторів коду, екранних форм та інших типів інструментальних засобів;

переведення на нову систему експлуатації – ця фаза включає проведення користувачами приймальних випробувань, встановлення системи й навчання користувачів.

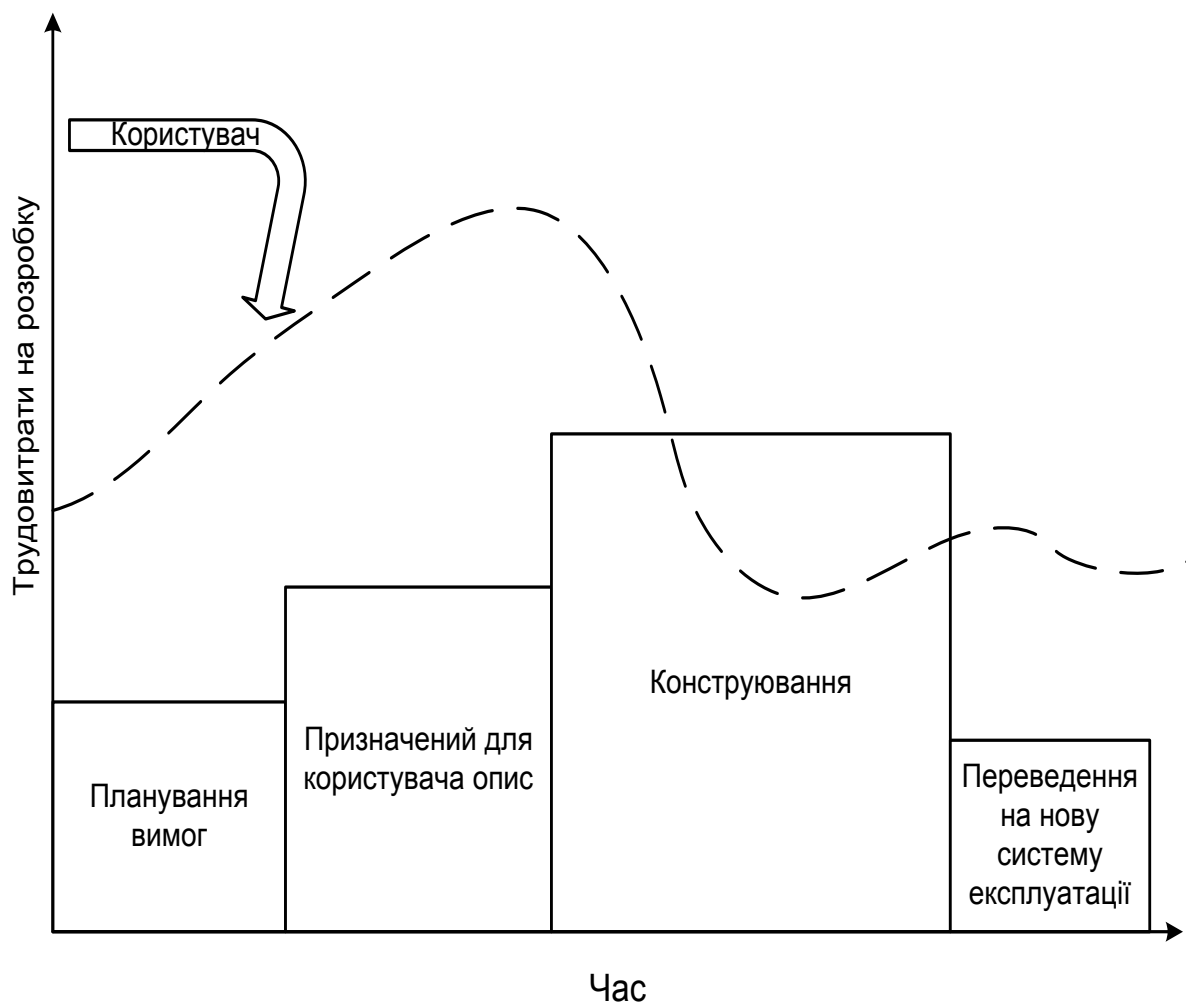


Рис. 21. **Модель швидкої розробки додатків**

Переваги моделі RAD:

час циклу розробки для всього проекту можна скоротити завдяки використанню потужних інструментальних засобів;

потрібна менша кількість фахівців, оскільки розробка системи виконується зусиллями команди, інформованої в предметній області;
існує можливість здійснити швидкий початковий перегляд продукту;
завдяки скороченому часу циклу і вдосконаленій технології, а також меншій кількості задіяних у процесі розробників зменшуються витрати;
залучення замовника на постійній основі зводить до мінімуму ризик того, що він не буде задоволений розробленим продуктом;
основна увага переноситься з документації на код, при цьому справедливий принцип "отримуєте те, що бачите" (What you see is what you get, WYSIWYG);
у моделі повторно використовуються компоненти вже існуючих програм.

Недоліки моделі RAD:

якщо користувачі не можуть постійно брати участі в процесі розробки впродовж всього життєвого циклу, це може негативно позначитися на кінцевому продукті;

використання моделі може виявитися невдалим у випадку, якщо відсутні придатні для повторного використання компоненти;

для реалізації моделі потрібні розробники й замовники, які готові до швидкого виконання дій незважаючи на жорсткі часові обмеження.

Сфера застосування моделі RAD:

у системах, які підлягають моделюванню (заснованих на використанні компонентних об'єктів), а також в масштабованих системах;

у системах, вимоги для яких достатньою мірою добре відомі;

у випадках, коли кінцевий користувач може брати участь у процесі розробки впродовж усього життєвого циклу;

коли команді, що працює над проектом, знайома предметна область;

у процесі виконання проектів, розробка яких повинна бути виконана в скорочені терміни, як правило, не більш ніж за два місяці;

коли можна отримати придатні до повторного використання частини програмних продуктів;

у системах, які призначені для концептуальної перевірки, є критичними або мають невеликий розмір;

у системах, у яких не потрібні досягнення високої продуктивності, за невисокого ступеня технічних ризиків;

в інформаційних системах.

9.3.6. Інкрементна (покрокова) модель

Інкрементна розробка є процесом поетапної реалізації всієї системи і поетапного нарощування функціональних можливостей. Цей підхід дозволяє зменшити витрати, завдані до моменту досягнення рівня вихідної продуктивності. За допомогою цієї моделі прискорюється процес створення системи, яка функціонує. Цьому сприяє використовуваний принцип компонування зі стандартних блоків, завдяки якому забезпечується контроль над процесом розробки вимог, що змінюються.

Інкрементна модель реалізується таким чином (рис. 22).

Перший крок. Необхідний повний, наперед сформульований набір вимог, які діляться за певною ознакою на частини.

Другий крок. Вибирається перша група вимог і виконується повний прохід по каскадній моделі. Після того, як перший варіант системи, що виконує першу групу вимог, зданий замовнику, розробники переходять до наступного кроку (другого інкременту) з розробки варіанта, який виконує другу групу вимог, і т. д.

Інкрементна модель описує процес, у ході виконання якого першорядна увага приділяється системним вимогам, а потім – їх реалізації в групах розробників. Як правило, з часом інкременти зменшуються і реалізують щоразу меншу кількість вимог. Кожна наступна версія системи додає до попередньої певні функціональні можливості доти, поки не будуть реалізовані всі заплановані можливості. У цьому випадку можна зменшити витрати, контролювати вплив вимог, що змінюються, і прискорити створення системи, яка функціонує завдяки використанню методу компонування зі стандартних блоків.

Передбачається, що на ранніх етапах життєвого циклу (планування, аналізу й розробки проекту) виконується конструювання системи в цілому. На цих етапах визначаються інкременти, що відносяться до них, і функції. Кожен інкремент проходить потім решту фаз життєвого циклу: кодування, тестування і постачання.

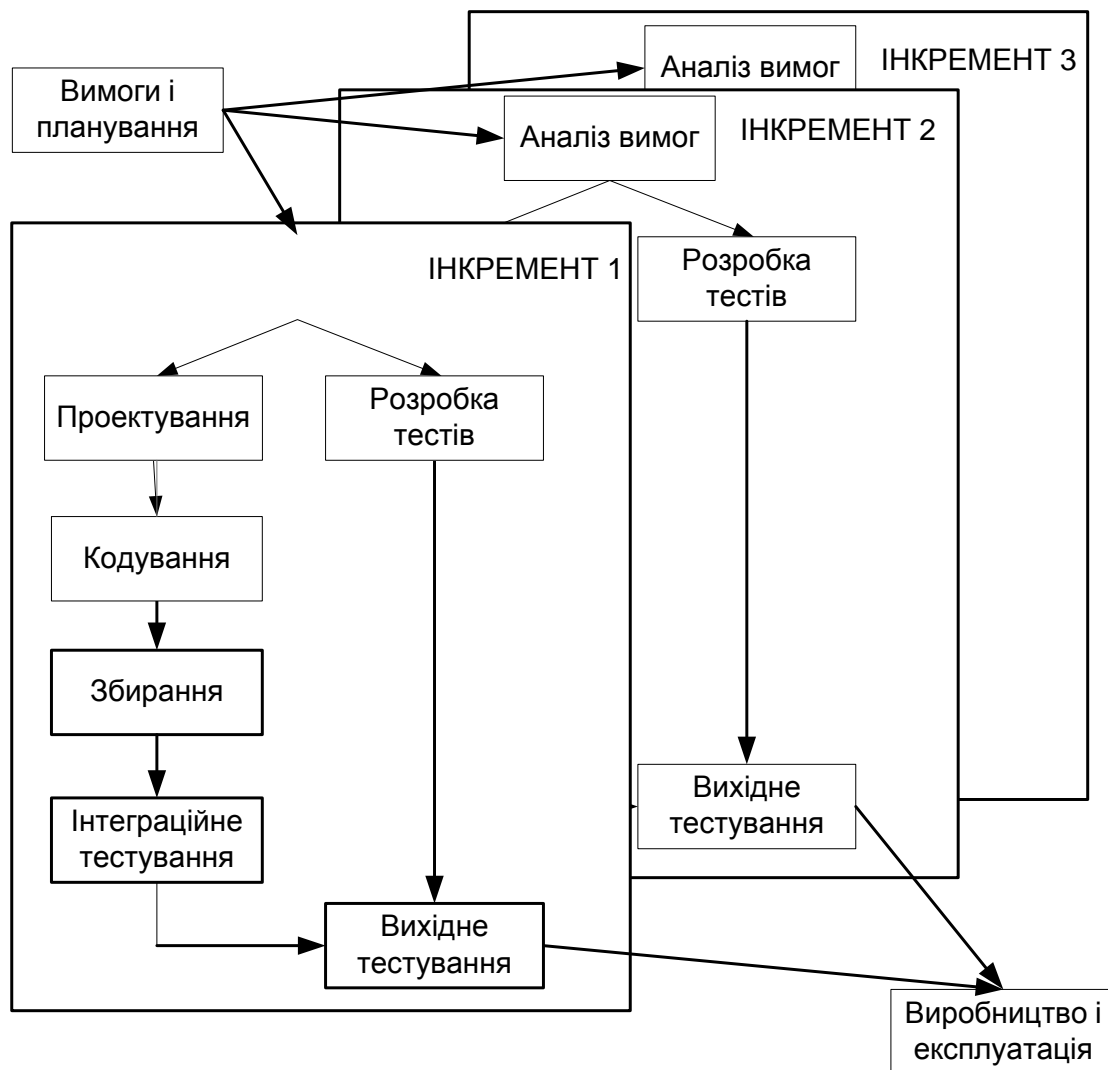


Рис. 22. Інкрементна модель

Спочатку виконується конструювання, тестування і реалізація набору функцій, які формують основу продукту, або вимог першорядного значення, що відіграють основну роль для успішного виконання проекту або знижують ступінь ризику. Наступні ітерації поширюються на ядро системи, поступово покращуючи її функціональні можливості або робочу характеристику. Додавання функцій здійснюється за допомогою виконання суттєвих інкрементів з метою комплексного задоволення потреб користувача. Кожна додаткова функція атестується відповідно до цілого набору вимог.

Особливістю інкрементної моделі є розробка приймальних тестів на етапі аналізу вимог, що спрощує приймання варіанта замовником і встановлює чіткі цілі розробки чергового варіанта системи. Інкрементна модель особливо ефективна в разі, коли завдання розбивається на декілька відносно незалежних підзавдань (розробка підсистем

"Персонал", "Касовий зал", "Склад", "Постачальники"). Крім того, інкрементна модель для внутрішньої ітерації може використовувати не лише каскадну, але й інші типи моделей – спіральну, V-подібну, що дозволяє знизити витрати і ризики в процесі розробки системи.

Лінійна послідовність виконання інкрементів може бути розподілена згідно з календарним графіком, причому внаслідок виконання кожної послідовності може створюватися результативний інкремент програмного продукту.

Переваги інкрементної моделі:

не потрібно наперед витратити кошти, необхідні для розробки всього проекту, оскільки виконуються розробка й реалізація основної функції з групи високого ризику;

внаслідок виконання кожного інкременту виходить готовий продукт; виникла проблема розбивається на керовані частини, а це запобігає формуванню громіздких переліків вимог, що висуваються перед командою розробників;

наприкінці постачання кожного інкременту існує можливість переглянути ризики, пов'язані з витратами і дотриманням встановленого графіка;

потреби клієнта краще піддаються управлінню, оскільки час розробки кожного інкременту дуже незначний;

замовники можуть розпізнавати найважливіші й найкорисніші функціональні можливості продукту на більш ранніх етапах розробки;

ризик розподіляється на декілька менших за розміром інкрементів і не зосереджений в одному великому проекті розробки;

вимоги стабілізуються (за допомогою включення в процес користувачів) на момент створення певного інкременту, оскільки вимоги, що не є особливо важливими, відкладають до моменту створення наступних інкрементів;

поліпшується розуміння вимог для пізніших інкрементів, що забезпечується завдяки можливості користувача отримати уявлення про раніше отримані інкременти на практичному рівні.

Недоліки інкрементної моделі:

у моделі не передбачені ітерації в рамках кожного інкременту;

визначення повної функціональної системи повинне здійснюватися на початку життєвого циклу, щоб забезпечити визначення інкрементів;

оскільки створення деяких модулів буде завершено значно раніше за інші, виникає необхідність у чітко визначених інтерфейсах;

формальний критичний аналіз і перевірку набагато важче виконати для інкрементів, ніж для системи в цілому;

може виникнути тенденція до відкладання вирішення важких проблем на майбутнє з метою продемонструвати керівництву успіх, досягнутий на ранніх етапах розробки.

Сфера застосування інкрементної моделі:

якщо більшість вимог можна сформулювати заздалегідь, але їх поява очікується через певний період часу;

якщо ринкове вікно надто вузьке і є потреба в швидкому постачанні на ринок продукту, що має функціональні базові властивості;

для проектів, на виконання яких передбачений великий період часу розробки, як правило, один рік;

під час розробки програм, пов'язаних із низьким або середнім ступенем ризику;

у процесі виконання проекту із застосуванням нової технології, що дозволяє користувачеві адаптуватися до системи шляхом виконання незначних інкрементних кроків, без різкого переходу до застосування основ-ного нового продукту;

коли однопрохідна розробка системи пов'язана з великим ступенем ризику.

9.3.7. Спіральна модель

Звичайно розробка ПЗ має циклічний характер, коли після виконання певних фаз доводиться повертатися до попередніх. Можна вказати дві основні причини таких повернень:

помилки розробників, допущені на ранніх стадіях і виявлені на пізніших стадіях: помилки аналізу вимог, проектування, кодування, що виявляються, як правило, на стадії тестування;

помилки замовників, що викликають зміну вимог у процесі розробки.

У спіральній моделі, вперше представленій Баррі Боемом у 1988 році, враховані недоліки каскадної моделі. Каскадна модель не здатна адекватно вирішувати проблеми внесення змін. У ній передбачається відносно стандартна і впорядкована послідовність стадій розробки, але в ній не передбачене використання методів прискореного прототипування або мов високого рівня.

Спіральна модель втілює в собі переваги каскадної моделі. При цьому в неї також включені:

аналіз ризиків, управління ними, а також процеси підтримки й управління;

розробка програмного продукту під час використання методу прототипування або швидкої розробки додатків за допомогою застосування мов програмування і розвинених засобів розробки.

Спіральна модель життєвого циклу ПЗ реалізується таким чином (рис. 23):

розробка варіантів продукту подається як набір циклів спіралі, що розкручується;

кожному циклу спіралі відповідає така ж кількість стадій, як і в каскадній моделі. При цьому початкові стадії, пов'язані з аналізом і плануванням вимог, наведені детальніше із додаванням нових елементів.

У кожному циклі моделі виділяються чотири базові фази:

визначення цілей, альтернативних варіантів і обмежень;

оцінка альтернативних варіантів, ідентифікація і вирішення ризиків;

розробка продукту наступного рівня;

планування наступної фази.

До початку розробки ПЗ є декілька повних циклів аналізу вимог і проектування. Кількість циклів моделі (як у частині аналізу і проектування, так і в частині реалізації) не обмежена і визначається складністю та обсягом завдання. У моделі передбачається повернення на залишені варіанти в разі зміни вартості ризиків.

Перший цикл – **створення бачення продукту**. Визначаються загальні цілі, встановлюються попередні обмеження, визначаються можливі альтернативи підходів до вирішення завдання. Далі проводиться оцінка підходів, встановлюються їх ризики. На кроці розробки створюється концепція (бачення) продукту і шляхів її створення.

Другий цикл – **аналіз вимог**. Починається з планування вимог і деталей ЖЦ продукту для оцінки витрат. На фазі визначення цілей встановлюються альтернативні варіанти вимог, пов'язані з ранжуванням вимог за важливістю й вартістю їх виконання. На фазі оцінки встановлюються ризики варіантів вимог. На фазі розробки –

специфікація вимог (із зазначенням ризиків і вартості), готується демоверсія ПЗ для аналізу вимог замовником.

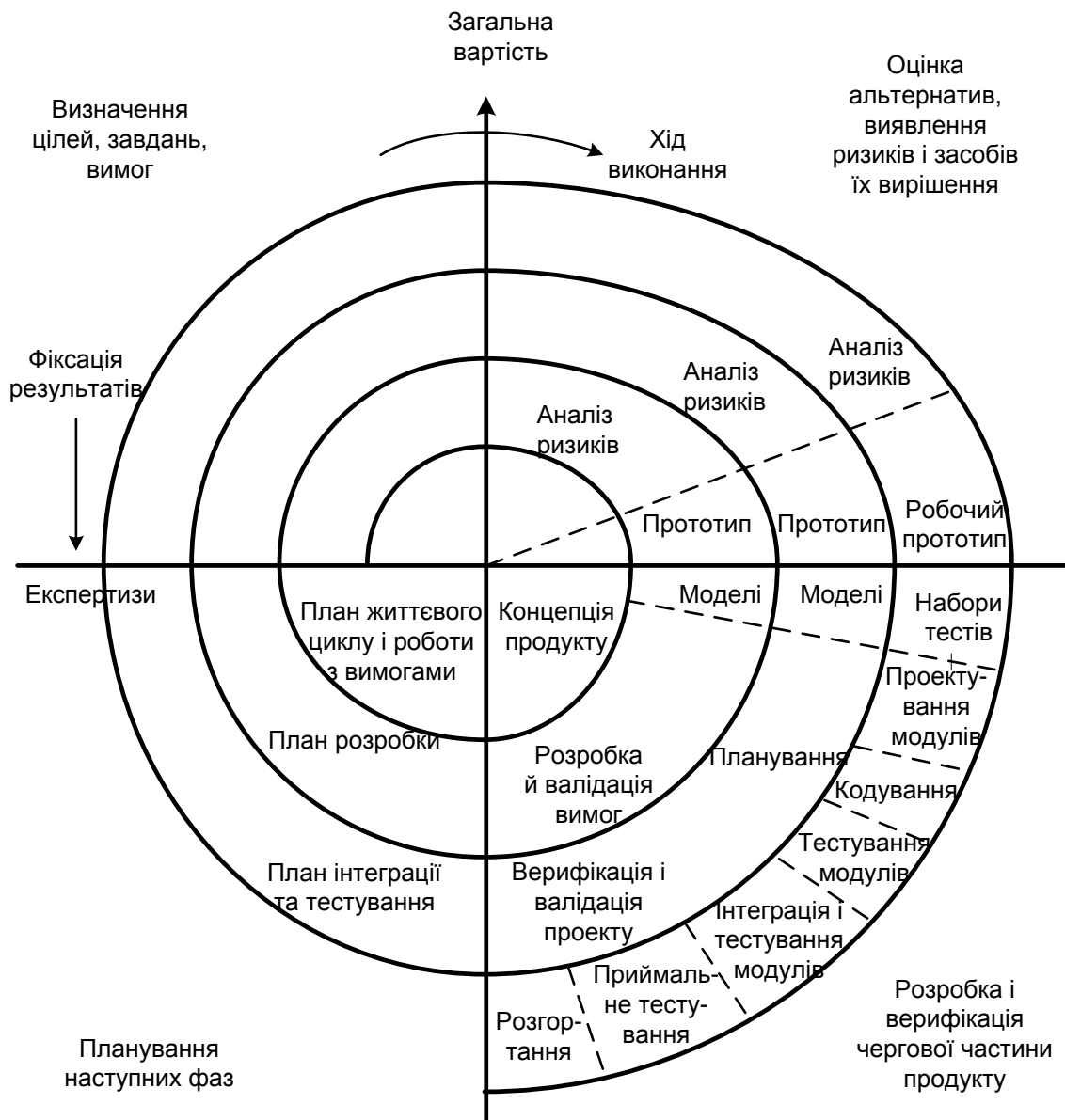


Рис. 23. Спиральна модель

Третій цикл – **розробка проекту**. Починається з планування розробки. На фазі визначення цілей встановлюються обмеження проекту (за термінами, обсягом фінансування, ресурсами і т. д.), визначаються альтернативи проектування, пов'язані з альтернативами вимог, використовуваними технологіями проектування, залученням субпідрядників тощо. На фазі оцінки альтернатив встановлюються ризики варіантів і здійснюється вибір варіанта для подальшої реалізації. На фазі

розробки виконується проектування і створюється демоверсія, що відбиває основні проектні рішення.

Четвертий цикл – **реалізація ПЗ**. Починається з планування реалізацій. Альтернативними варіантами реалізації можуть бути використовувані технології реалізації, ресурси, що залучаються. Оцінка альтернатив і пов'язаних із ними ризиків у цьому циклі визначається ступенем "відпрацьованості" технологій і "якістю" наявних ресурсів. Фаза розробки виконується за каскадною моделлю, виходом якої є діючий варіант (прототип) продукту.

Основні принципи спіральної моделі можна сформулювати таким чином:

- розробка варіантів продукту, відповідних різним варіантам вимог, з можливістю повернутися до попередніх варіантів;

- створення прототипів ПЗ як засобів спілкування із замовником для уточнення і виявлення вимог;

- планування наступних варіантів з оцінкою альтернатив і аналізом ризиків, пов'язаних з переходом до наступного варіанта;

- перехід до розробки наступного варіанта до завершення попереднього у разі, коли ризик завершення чергового варіанта (прототипу) стає невиправдано високим;

- використання каскадної моделі як схеми розробки чергового варіанта;

- активне залучення замовника до роботи над проектом. Замовник бере участь в оцінці чергового прототипу ПЗ, уточненні вимог у разі переходу до наступного, оцінці запропонованих альтернатив чергового варіанта й оцінці ризиків.

Переваги спіральної моделі:

- спіральна модель дозволяє користувачеві побачити систему на ранніх етапах за рахунок швидкого прототипування;

- забезпечується визначення нездоланих ризиків без додаткових витрат;

- модель дозволяє користувачам активно брати участь у плануванні, аналізі ризиків, розробці й оцінюванні;

- модель забезпечує розподіл великого потенційного обсягу роботи з розробки продукту на невеликі частини, в яких спочатку реалізуються вирішальні функції з високим ступенем ризику, тому в разі потреби стає можливим припинити роботу над проектом і зменшуються витрати;

у моделі передбачена можливість гнучкого проектування, оскільки в ній втілені переваги каскадної моделі, а також дозволені ітерації для всіх фаз цієї ж моделі;

реалізовані переваги інкрементної моделі, а саме випуск інкрементів, скорочення графіка за допомогою перекривання інкрементів, розсортованих за версіями, і незмінність ресурсів при поступовому зростанні системи;

зворотний зв'язок у напрямі від користувачів до розробників виконується з високою частотою і на ранніх етапах моделі, що забезпечує створення потрібного продукту високої якості;

відбувається вдосконалення адміністративного управління над процесом забезпечення якості, правильністю виконання процесу розробки, витратами, дотриманням графіка і кадровим забезпеченням, що досягається шляхом виконання огляду в кінці кожної ітерації;

підвищується продуктивність завдяки використанню придатних для повторного використання властивостей;

підвищується ймовірність передбаченої поведінки системи за допомогою уточнення поставлених цілей;

у разі використання спіральної моделі не потрібно розподіляти заздалегідь усі необхідні для виконання проекту фінансові ресурси;

можна виконувати часту оцінку сукупних витрат, а зменшення ризиків пов'язане з витратами.

Недоліки спіральної моделі:

складність аналізу й оцінки ризиків під час вибору варіантів. Якщо проект має низький ступінь ризику або невеликі розміри, модель може виявитися дорогою;

складність структури моделі, що може ускладнити її застосування розробниками, менеджерами й замовниками;

складність підтримки версій продукту (зберігання версій, повернення до ранніх версій, комбінація версій, документування версій);

нескінченність моделі – після кожної створеної версії замовник може висувати нові вимоги, які приводять до необхідності наступного циклу розробки.

Сфера застосування спіральної моделі:

коли користувачі не впевнені у своїх потребах;

коли вимоги надто складні й можуть змінюватися в процесі виконання проекту і необхідне прототипування для аналізу та оцінки вимог;

у процесі розробки нової функції або нової серії продуктів;

коли досягнення успіху не гарантоване і необхідна оцінка ризику продовження проекту;

коли проект є складним, дорогим і обґрунтування його фінансування можливе лише в процесі його виконання;

коли мова йде про застосування нових технологій, що пов'язане з ризиком їх освоєння і досягнення очікуваного результату;

у ході виконання дуже великих проектів, які через обмеженість ресурсів можна здійснювати лише частинами;

коли очікуються суттєві зміни, наприклад, у процесі вивчення або дослідницької роботи;

для організацій, які не можуть собі дозволити виділити наперед всі необхідні для виконання проекту грошові кошти, і коли в процесі розробки відсутня фінансова підтримка;

коли переваги розробки неможливо точно визначити, а досягнення успіху не гарантоване;

у процесі розробки систем, що вимагають великого обсягу обчислень, таких як системи, що забезпечують ухвалення рішень, у ході виконання бізнес-проектів, а також проектів у галузі оборони, інжинірингу і т. д.

9.4. Моделі життєвого циклу програмного продукту на основі індустріальних технологій

9.4.1. Важкі та гнучкі технології проектування

На сьогодні широкого застосування набувають так звані індустріальні технології створення програмного продукту. Ці технології були розроблені фірмами, що накопичили великий досвід у створенні ПЗ. Технології представлені описами принципів, методів, використовуваних процесів і операцій. Такі технології, як правило, підтримуються набором CASE-засобів, охоплюють усі етапи життєвого циклу продукту й успішно застосовуються для вирішення практичних завдань.

Індустріальні технології створення програмного продукту можна розділити на два види:

із важкими процесами розробки;

із гнучкими (agile) процесами розробки.

Важкі технології детально описуються і передбачають підтримку розробки початкового коду ПЗ великою кількістю допоміжних дій. Прикладами подібних дій є розробка планів, технічних завдань (специфікацій вимог), проектних моделей, проектної документації і т. д.

Основна мета такого процесу – відокремити успішні практики розробки і супроводу ПЗ від конкретних людей, що вміють їх застосовувати. Численні допоміжні дії дозволяють зробити можливим успішне вирішення завдань з конструювання й підтримки складних систем за допомогою наявних працівників, що не обов'язково є високими професіоналами.

Для досягнення цього виконується ієрархічний покроковий детальний опис дій, що здійснюються в тій чи іншій ситуації, щоб можна було навчити звичайного працівника діяти аналогічним чином. У ході проекту створюється багато проміжних документів, що дозволяють розробникам послідовно розбивати завдання, що постають перед ними, на простіші. Ці ж документи слугують для перевірки правильності рішень, що ухвалюються на кожному кроці, а також відстеження загального ходу робіт і уточнення оцінок ресурсів, необхідних для отримання бажаних результатів.

До найбільш відомих моделей важкої розробки відноситься раціональний уніфікований процес (Rational Unified Process, RUP), методологія Microsoft Solution Framework (MSF) тощо.

Гнучкі технології, які ще називають **живими** або **легкими**, уникають фіксації чітких схем дій, щоб забезпечити велику гнучкість у кожному конкретному проекті, а також виступають проти розробки додаткових документів, що не роблять безпосереднього внеску в отримання готової дієвої програми.

Гнучкі методи з'явилися як протест проти надмірної бюрократизації розробки ПЗ, великої кількості побічних, таких, що не є необхідними для отримання кінцевого результату, документів, які доводиться оформляти у процесі проведення проекту відповідно до більшості важких процесів, додаткової роботи з підтримки фіксованого процесу організації. Велика частина таких робіт і документів не має прямого відношення до розробки ПЗ і забезпечення його якості, а призначена для дотримання формальних пунктів контрактів на розробку, отримання і підтвердження сертифікатів на відповідність різним стандартам.

Гнучкі методи дозволяють велику частину зусиль розробників зосередити власне на завданнях розробки і задоволення реальних потреб користувачів. Відсутність безлічі документів і необхідності підтримувати їх в зв'язному стані дозволяє швидко і якісно реагувати на

зміни у вимогах і в оточенні, в якому доведеться працювати майбутній програмі.

До моделей ЖЦ ПЗ, що підтримують Agile-технології, належать метод екстремального програмування (eXtreme Programming, XP), методи Crystal Clear, Feature-Driven Development (розробка, керована функціями системи), DSDM (Dynamic Systems Development Method, метод розробки динамічних систем) та ін. Основні принципи гнучкої розробки ПЗ зафіксовані в Маніфесті проекту **Agile Developer**, що з'явився у 2000 році.

Загальні риси гнучких методологій наступні:

ітеративний підхід, за якого детально планується лише обмежений обсяг робіт, пов'язаний із випуском чергового релізу;

неформальний підхід, згідно з яким, якщо проблему можна вирішити в розмові, то краще саме так і зробити. Причому оформляти ухвалені рішення у вигляді паперового або електронного документа потрібно лише тоді, коли без цього неможливо обійтися;

обмежений опис, відповідно до якого всі гнучкі методології використовують досить обмежений набір документів, моделей і робіт для опису процесу розробки. Це робить їх досить простими (принаймні, на перший погляд) для впровадження. Хоча часто ця простота досягається лише за рахунок того, що багато, безумовно, необхідних робіт на зразок управління конфігураціями тільки згадуються, а не описуються в методології.

Таким чином, як важкі, так і гнучкі технології підтримують ітеративні процеси розробки ПЗ. Але кожна з цих технологій має свої особливості організації моделі життєвого циклу створення продукту відповідно до природи розробки програмного забезпечення.

9.4.2. Методологія RUP

Методологія **Rational Unified Process (RUP)** є розробкою компанії Rational. Ця компанія довгий час успішно займається створенням CASE-засобів, використовуваних на різних етапах життєвого циклу продукту від аналізу до тестування і документування. Технологія RUP універсальна, масштабована і налаштовується на застосування в конкретних умовах. Модель життєвого циклу RUP є досить складною ітеративно-інкрементною моделлю, що детально відпрацьована, з елементами каскадної моделі.

Основними **характеристиками** RUP є наступні:

використання кращих практичних методів, що успішно зарекомендували себе в багатьох проектах розробки ПЗ у всьому світі;

чітко визначений технологічний процес, що описує структуру життєвого циклу проекту, ролі й відповідальності окремих виконавців, виконувани ними завдання і використовувани в процесі розробки моделі, звіти і т. д.;

готовий продукт, що подається у вигляді веб-сайта, який містить усі необхідні моделі й документи із описом процесу.

Підхід до розробки. RUP пропонує розробникам не жорсткі правила, що регламентують виконання всіх дій у ході розробки, а набір досить гнучких методів і підходів, із яких розробник може вибрати те, що понад усе відповідає його завданням і особливостям проекту.

Основними **принципами** RUP є:

ітераційна розробка;

управління процесом на основі прецедентів використання;

орієнтація на архітектуру.

Слід розглянути ці принципи детальніше.

Ітерація – це закінчений цикл розробки, що приводить до випуску кінцевого продукту або певної його скороченої версії, яка розширюється від ітерації до ітерації, щоб зрештою стати закінченою системою (рис. 24). У разі ітераційного підходу кожна з фаз процесу розробки ПЗ складається з ітерацій, метою яких є послідовне осмислення проблем, що постають, нарощування ефективності рішень і зниження ризиків потенційних помилок у проекті. На виході кожної ітерації створюється закінчена версія дієвого програмного продукту. За такого підходу можна гнучко враховувати нові вимоги або проводити тактичні зміни в ділових цілях, виявляти проблеми і вирішувати їх на більш ранніх етапах розробки, що пов'язане з меншими витратами.

Управління на основі прецедентів використання означає, що як метод опису функціональних вимог до системи, а також як природна одиниця для подальшого планування й оцінки виконання робіт застосовуються сценарії використання. Сценарії використання дозволяють легко виявляти реальні потреби майбутніх користувачів системи і відстежувати повноту опису цих вимог.

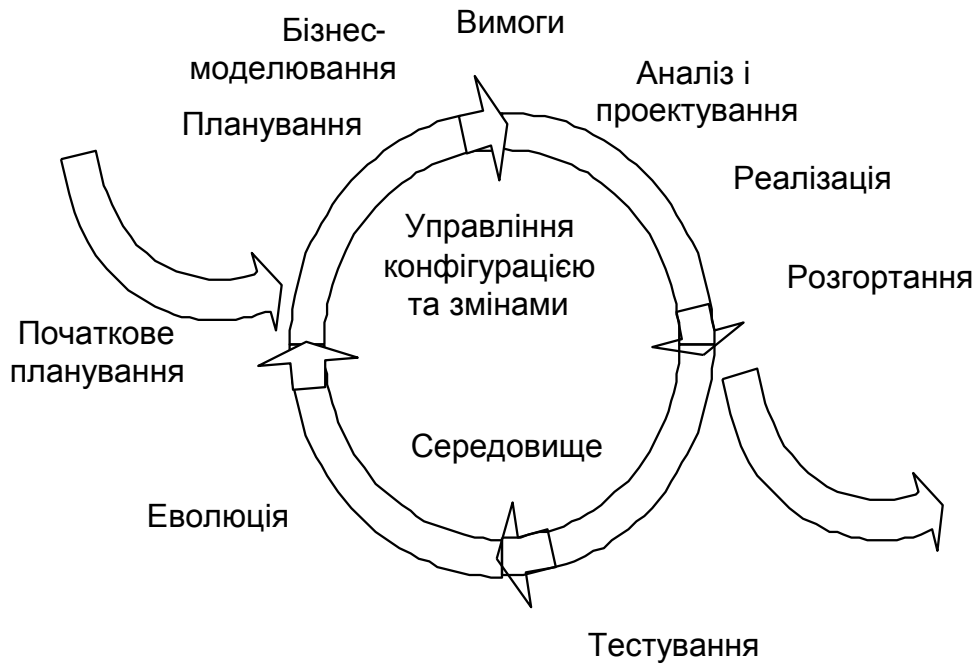


Рис. 24. Ітерація ЖЦ RUP

Вони гарантують виконання вимог замовника до ПЗ. Крім того, використання завершених сценаріїв як одиниця вимірювання прогресу допомагає уникнути неадекватної оцінки ступеня виконання проекту виконавцем.

Орієнтація на архітектуру передбачає розробку, реалізацію і тестування архітектури на більш ранніх стадіях виконання проекту. Такий підхід дозволяє усувати найнебезпечніші ризики, пов'язані з архітектурою, на ранніх стадіях розробки. Завдяки йому вдається уникнути суттєвих переробок в останню мить, якщо раптом з'ясується, що вибране рішення не забезпечує, наприклад, виконання вимог до безпеки системи.

RUP створювався за методикою, використовуюваною в процесі проектування ПЗ. Зокрема, моделювання проводилося за допомогою Software Process Engineering Metamodel (SPEM) – стандарту моделювання процесів, заснованого на Unified Modeling Language (UML). У моделі ЖЦ RUP є два виміри (рис. 25): динамічне і статичне.

Дисципліни (процеси)

Бізнес-моделювання
Визначення вимог
Аналіз і проектування
Реалізація
Тестування
Розгортання
Управління конфігураціями і змінами
Управління проектом
Управління середовищем проекту

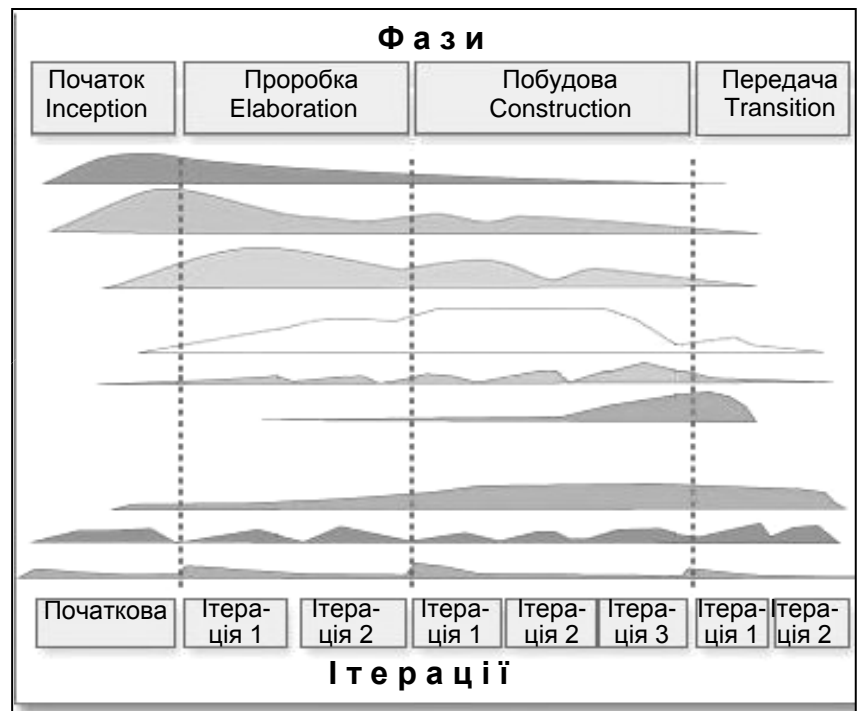


Рис. 25. Модель ЖЦ RUP

Динамічна структура. Горизонтальний вимір є динамічною структурою або часовим виміром процесу. Він показує, як процес, виражений у формі циклів, фаз, ітерацій і віх, розгортається в ході життєвого циклу проекту.

Динамічна структура RUP складається з чотирьох фаз:

фаза початку проекту (Inception). Визначаються основні цілі проекту, бюджет проекту, основні засоби його виконання – технології, інструменти, ключовий персонал, складаються попередні плани проекту. Основна мета цієї фази – досягти компромісу між усіма зацікавленими особами щодо завдань проекту;

фаза проробки (Elaboration). Основна мета цієї фази – на базі основних, найбільш суттєвих вимог розробити стабільну базову архітектуру продукту, яка дозволяє вирішувати поставлені перед системою завдання і надалі використовується як основа розробки системи;

фаза побудови (Construction). Основна мета цієї фази – детальне прояснення вимог і розробка системи, що задовольняє їх, на основі спроектованої раніше архітектури;

фаза передачі (Transition). Мета фази – зробити систему цілком доступною кінцевим користувачам. Тут відбувається остаточне

розгортання системи в її робочому середовищі, підгонка дрібних деталей під потреби користувачів.

Фази можуть підрозділятися на ітерації. Перехід з фази на фазу можливий лише після виконання завдань фази і є контрольною точкою – віхою (milestone) – процесу.

Статична структура. Вертикальний вимір є статичною структурою процесу. Він описує, як елементи процесу (завдання, дисципліни, артефакти і ролі) логічно групуються в дисципліни або робочі процеси (workflows). Для опису осмисленої послідовності виконання робіт і завдань використовуються робочі процеси. Вони описують, хто, що, як і коли виконує в процесі. У RUP входять 6 основних дисциплін і 3 допоміжних.

До основних дисциплін належать:

моделювання предметної області (бізнес-моделювання, Business Modeling). Цілі цієї діяльності – зрозуміти бізнес-контекст, в якому повинна буде працювати система (і переконатися, що всі зацікавлені особи розуміють його однаково), зрозуміти можливі проблеми, оцінити їх можливі розв'язання і наслідки для бізнесу організації, в якій працюватиме система;

визначення вимог (Requirements). Цілі – зрозуміти, що повинна робити система, визначити межі системи і основу для планування проекту і оцінок ресурсовитрат у ньому;

аналіз і проектування (Analysis and Design). Розробка архітектури системи на основі ключових вимог, створення проектної моделі, наведеної у вигляді діаграм UML, що описують продукт з різних точок зору;

реалізація (Implementation). Розробка початкового коду, компонент системи, тестування та інтеграція компонент;

тестування (Test). Загальна оцінка дефектів продукту, його якості в цілому, оцінка ступеня відповідності початковим вимогам;

розгортання (Deployment). Цілі – розгорнути систему в її робочому оточенні і оцінити її працездатність.

До допоміжних дисциплін відносяться:

управління конфігураціями і змінами (Configuration and Change Management). Визначення елементів, що підлягають зберіганню і правил

побудови з них узгоджених конфігурацій, підтримка цілісності поточного стану системи, перевірка узгодженості змін, що вносяться;

управління проектом (Project Management). Включає планування, управління персоналом, забезпечення зв'язків з іншими зацікавленими особами, управління ризиками, відстеження поточного стану проекту;

середовище проекту (Environment). Настроювання процесу під конкретний проект, вибір і зміна технологій і інструментів, використовуваних у проекті.

На відміну від каскадного підходу, в RUP усі дисципліни виконуються практично на всіх фазах життєвого циклу ПС. Однак, залежно від фази, змінюються поточні цілі проекту і співвідношення між обсягами робіт, відповідних різним дисциплінам.

Для забезпечення інструментальної підтримки всіх процесів життєвого циклу розробки і супроводу ПЗ RUP пропонує спеціалізовані інструментальні засоби IBM Rational (рис. 26).

RequisitePro – дозволяє створювати і оновлювати керовані ним вимоги безпосередньо в програмі або за допомогою редактора Microsoft Word. Для зміни вимог досить відредагувати вміст документа, при цьому сам засіб допоможе відновити, проаналізувати й об'єднати необхідні вимоги.

Software Modeler – засіб, що замінив компоненту Rational Rose® XDE, яка раніше використовувалася. Побудоване на платформі Eclipse, вона успішно використовувалося в сімействі продуктів IBM WebSphere® як основа для інтегрованого середовища розробки (IDE) WebSphere Studio Application Developer. Rational Software Modeler надає користувачу інтерфейс для моделювання додатка до початку написання програмного коду.

Application Developer – надає середовище IDE для перетворення моделей в програмний код і фінальні додатки. Rational Application Developer також забезпечує користувачів інтегрованим середовищем для тестування готового додатка.

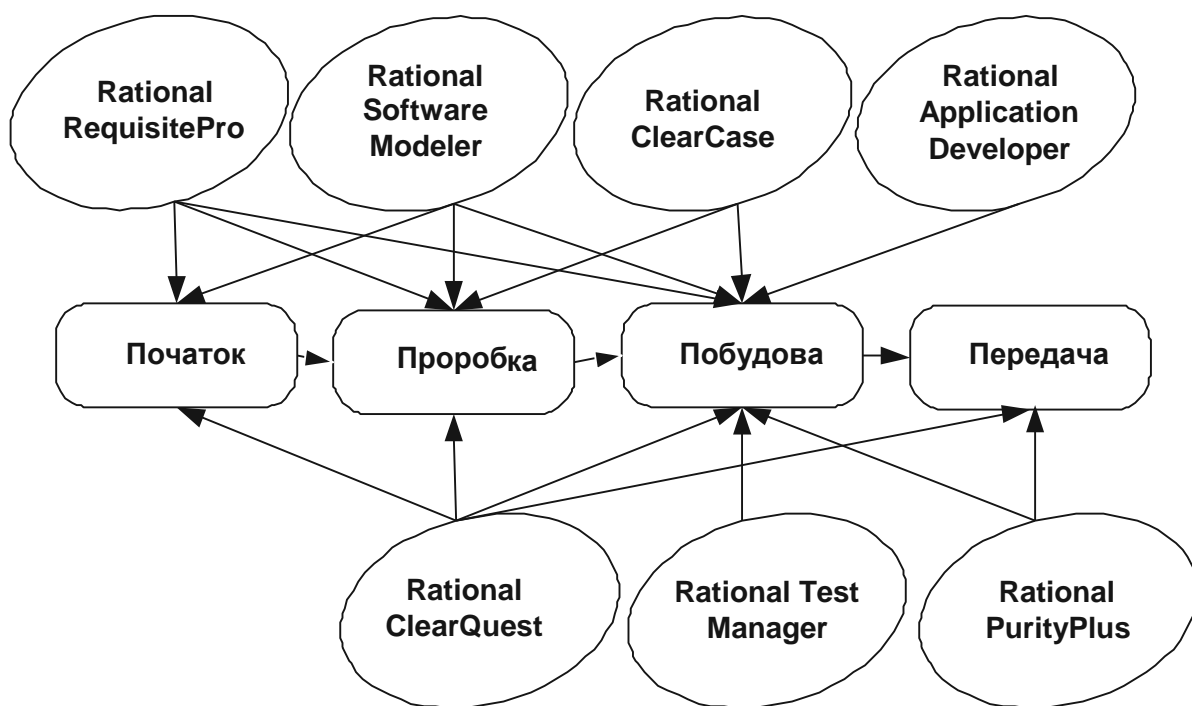


Рис. 26. Підтримка процесів ЖЦ інструментальними засобами RUP

ClearCase – забезпечує управління конфігурацією бази коду і моделей. WebSphere Studio Application Developer і Rational Rose XDE, взаємодіючи з Rational ClearCase, здатні забезпечити всю команду розробників механізмами управління всіма аспектами процесу розробки.

ClearQuest – полегшує відстеження дефектів та управління операціями. Відстеження дефектів дозволяє виявити помилки і знайти їх первинну причину. Механізм управління операціями дає можливість відстежувати операції над різними компонентами проекту, включаючи програм-ний код, моделі та іншу інформацію. Інтеграція ClearQuest з іншими додатками Rational дозволяє відстежувати дефекти й операції в широкому діапазоні компонентів і вимог, перевіряючи інформацію в будь-якій точці проекту.

PurifyPlus – забезпечує поточний аналіз додатка, що дозволяє простежити, як він використовує ресурси пам'яті. "Витік" пам'яті та інші проблеми такого роду є одним з найпоширеніших дефектів в додатках, за винятком друкарських помилок і помилок в програмному коді, які повинні усуватися за допомогою тестування і спеціальних методик, використовуваних під час застосування інструментальних засобів Rational. Інтеграція цього інструменту з середовищем WebSphere Studio Application Developer забезпечує користувачів інформацією про елементи додатка, що викликають ті або інші проблеми.

TestManager – надає єдиний інтерфейс для тестування набору параметрів додатка, дозволяючи проводити це тестування в автоматичному режимі, що особливо важливо, якщо розробка виконується на різних платформах.

Інструментарій RUP достатньо повний. Це набір рекомендацій і прикладів для всіх стадій і фаз розробки програм. Хоча в основу цих рекомендацій покладений багаторічний досвід розробки програмних систем, не для кожного проекту RUP підходить на сто відсотків. Будь-який програмний проект по-своєму унікальний. Не можна бездумно копіювати чужий процес, створюючи артефакти, що мають незначну цінність. У багатьох невеликих організаціях з розробки програмного забезпечення, особливо в тих, які не мають власної могутньої системи розробки, RUP можна використовувати "як є", але він може бути уточнений, розширений і специфічно настроєний для максимального наближення до потреб організації-розробника.

9.4.3. Методологія Microsoft Solution Framework

Методологія **Microsoft Solution Framework (MSF)** є розробкою компанії Microsoft, призначеної для вирішення широкого кола завдань. Технологія масштабована, тобто налаштовується на вирішення завдань будь-якої складності колективом будь-якої чисельності. Методологія MSF схожа з методологією RUP: модель ЖЦ ІС включає фази та ітерації розробки, заснована на об'єктно-орієнтованому моделюванні. Однак модель MSF більшою мірою, ніж RUP, орієнтована на розробку бізнес-додатків, що відбите в її основних концепціях.

Одна з особливостей технології MSF полягає в тому, що вона орієнтована не просто на створення програмного продукту, що задовольняє перераховані вимоги, а на пошук вирішення проблем, що постають перед замовником. Відмінність полягає в тому, що перераховані замовником вимоги є проявами деяких глибших проблем, і неточність, неповнота, зміна вимог у процесі розробки – наслідок нерозуміння проблем. Тому в технології MSF велика увага приділяється аналізу проблем замовника і розробці варіантів системи для пошуку вирішення цих проблем.

Модель життєвого циклу MSF є певним **поєднанням каскадної і спіральної моделей**. У ній поєднується простота управління каскадної

моделі із гнучкістю спіральної. Перебіг життєвого циклу проекту має вигляд послідовності ітерацій на витках спіралі (рис. 27).

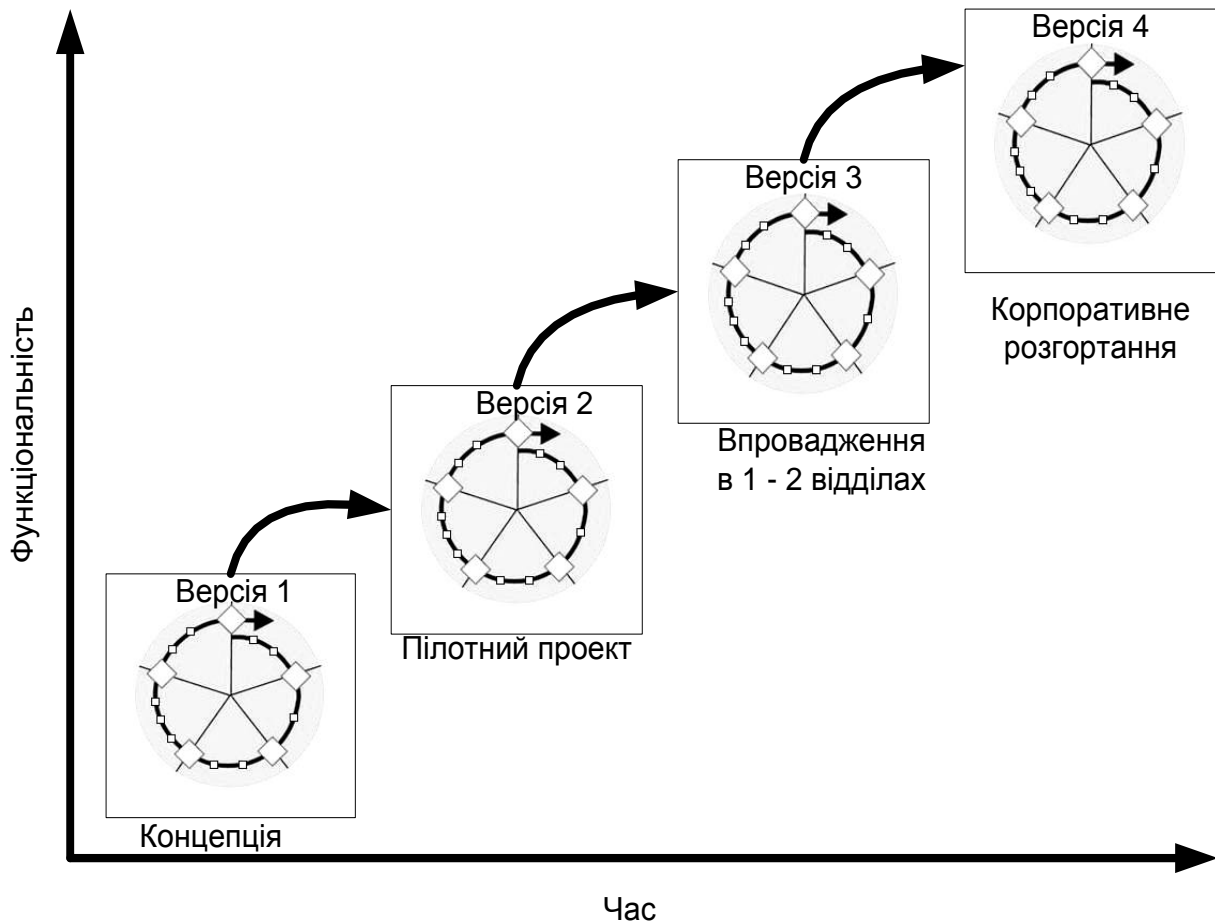


Рис. 27. Схеми моделі життєвого циклу Microsoft Solution Framework

Основні концепції і принципи моделі:

1. Єдине бачення проекту. Усі зацікавлені особи й учасники проекту повинні чітко уявляти кінцевий результат, тобто мати єдине бачення (Shared Vision) проекту. Іноді єдина мета проекту вступає в конфлікт з інтересами деяких його учасників або учасники проекту мають вигоду тільки від однієї підсистеми проекту, яка стосується їх безпосередньо. Наприклад, на останніх етапах проекту дуже важко пояснити комерційникам, навіщо потрібно вводити номер накладної на відпущений товар, тому що він йому не потрібен. Проте він потрібен бухгалтерії. Конфлікт виникає, тому що комерційники думають про автоматизацію лише завдань складу, а не підприємства в цілому. Тому в життєвому циклі MSF виділяється фаза для забезпечення єдиного бачення проекту.

2. Управління компромісами. У процесі управління проектом менеджерів необхідно знаходити компроміс між ресурсами проекту

(людськими, фінансовими і т. д.), календарним графіком (часом) і реальними можливостям і (рамками). Цю залежність можна відобразити за допомогою трикутника компромісів (рис. 28). Будь-яка зміна на одній зі сторін трикутника спричиняє зміни на двох інших сторонах.

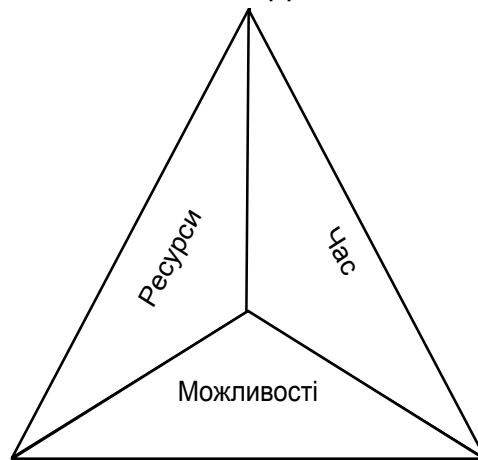


Рис. 28. Трикутник компромісів

Для ефективного досягнення компромісів у перебігу всього життєвого циклу проекту рекомендується на початкових етапах зафіксувати пріоритети чинників проекту (табл. 5):

один чинник **фіксується**, впливати на нього в процесі проектування буде практично неможливо;

другий чинник **узгоджується**, тобто матиме певний пріоритет під час знаходження компромісів;

третій чинник **приймається**, тобто буде прийнятий відповідно до перших двох.

3. Гнучкість до змін. MSF ґрунтується на принципі проектних умов, що змінюються. Для цього є ефективний інструментарій, що дозволяє адекватно та своєчасно реагувати на зміни вимог до проекту впродовж життєвого циклу.

Таблиця 5

Матриця компромісів

Пріоритет	Фіксується	Узгоджується	Приймається
Чинники			
Ресурси	✓		
Час		✓	
Можливості			✓

4. Урахування бізнес-пріоритетів. Необхідно зосередитися на тому результаті й вигоді, яку очікує отримати від програмного продукту його споживач. Якщо це проект, призначений для управління організацією, то ключовою точкою тут буде бізнес-віддача (business-value). Якщо це персональні програми, наприклад комп'ютерна гра, то ключовою точкою тут буде віддача у вигляді емоційного задоволення споживача.

5. Вільне спілкування. Для того, щоб усунути недовіру всередині групи розробників, методологія MSF не обмежує доступність до інформації серед членів групи проекту. Крім того, заохочується вільне спілкування всередині групи, що значно зменшує ризик нерозуміння, виникнення суперечностей і стимулює максимальний внесок усіх учасників проекту в загальну справу.

6. Створення базових версій. У першу версію продукту рекомендується вкладати тільки базову функціональність. Створення базової версії всіх проектних документів на більш ранніх етапах дає можливість усім членам проектної групи осмислити цілі проекту, свої завдання і розпочати роботу з мінімальними затримками. У наступних версіях продукту функціональність зростатиме. Для малих проектів може бути достатньо однієї ітерації. Рекомендується, однак, використовувати можливості версифіціювання як ефективного інструменту досягнення успіху.

Ітераційний підхід в життєвому циклі MSF виявляється в наступному: перегляд проектних артефактів, таких як вимоги, функціональність, плани і т. д., проводиться після кожної ітерації. Це дозволяє планувати можливості для наступних версій, враховувати досвід попередніх планів, забезпечуючи гнучкість і стійкість до вимог, що змінюються. За рахунок цього забезпечується відображення в документації всіх змін, пов'язаних з еволюцією проекту;

у перебігу однієї ітерації всі документи, включаючи програмний код, також розвиваються ітеративно. Наприклад, документ "Концепція проекту" (shared vision document) поширюється серед усіх учасників проекту у вигляді "підходів" (approaches) ще до його затвердження і потім ітеративно еволюціонує завдяки зворотному зв'язку між учасниками проекту до форми, що підлягає затвердженню. Усі зміни у вже прийняті документи в рамках

однієї ітерації (наприклад, затверджені вимоги до системи – ТЗ) вносяться на основі компромісів між членами проектної групи і відповідно до технології управління змінами. Часто ці зміни є сенс відкласти до наступної версії, щоб не допустити розростання проекту і не зірвати термінів здачі поточної версії.

У методології MSF рекомендується виконувати:

поточні збирання програмних компонент проекту за допомогою щоденних білдів (daily builds);

одночасну розробку і тестування програмних компонент для підвищення надійності програмного коду;

управління конфігураціями проекту (як ефективний інструмент управління змінами в проекті від запиту на зміну до включення його в базову версію);

жорсткий моніторинг і контроль версій програмного коду, документації, апаратних налаштувань та інших артефактів проекту.

Основні рекомендації для створення версій проекту:

у ході планування необхідно враховувати можливість створення версій проекту;

вибір пріоритетів створення версій проекту з урахуванням ризиків;

здійснення частих ітерацій розробки;

чітка регламентація процедур контролю змін у проекті;

нові версії створюються лише в тому випадку, якщо вони збільшують цінність рішення.

Підхід, заснований **на фазах і віхах** в життєвому циклі MSF, виявляється в наступному.

Модель життєвого циклу MSF орієнтована на віхи (milestones) – ключові точки проекту, що характеризують досягнення якого-небудь істотного результату. Результат у кожній вісі може бути оцінений і проаналізований. У моделі передбачається наявність основних віх (завершення головних фаз моделі) і проміжних, таких, що відбивають внутрішні етапи головних фаз (рис. 29).

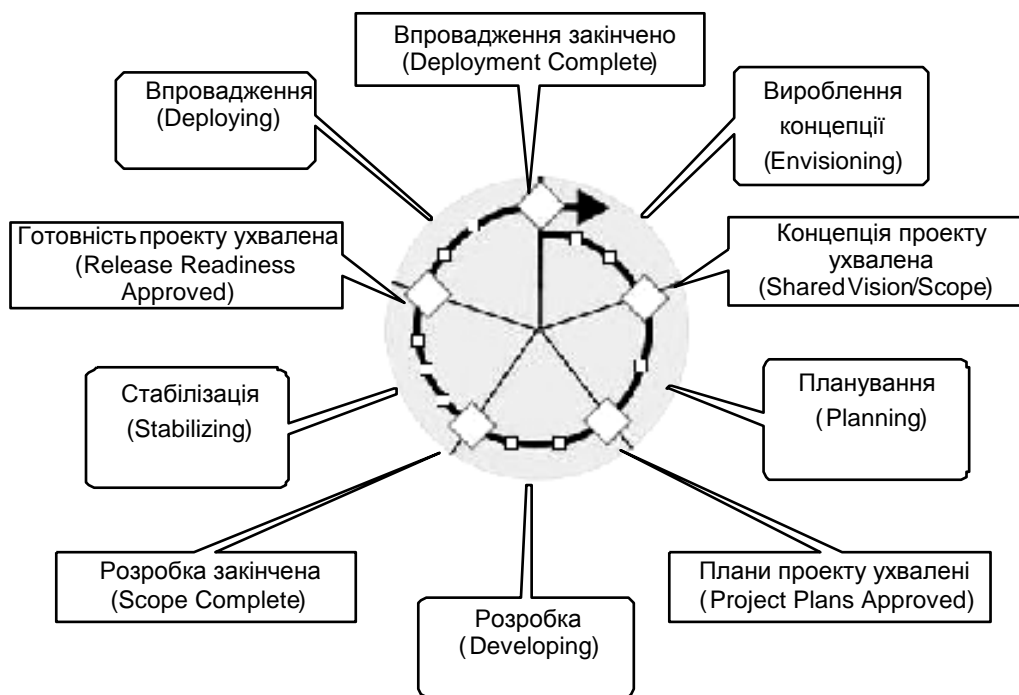


Рис. 29. Фази та віхи моделі життєвого циклу Microsoft Solution Framework

У рамках однієї ітерації життєвий цикл розбивається на п'ять фаз: вироблення концепції (загального бачення), планування, розробки, тестування (стабілізації), впровадження. Розглянемо детальніше ці фази.

Створення концепції (бачення) проекту – Envisioning. На цій фазі розв'язуються наступні основні завдання: оцінка наявної ситуації; визначення складу команди, структури проекту, бізнес-цілей, вимог і профілів користувачів; розробка концепції розв'язання і оцінки ризиків. Встановлюються дві проміжні віхи: "Організовано ядро команди" і "Створено бачення проекту".

Планування – Planning. Ця фаза включає планування і проектування продукту. На основі аналізу вимог розробляється проект і основні архітектурні рішення, функціональні специфікації системи, плани й календарні графіки, середовище розробки, тестування та пілотної експлуатації. Фаза складається з трьох етапів: концептуального, логічного і фізичного проектування. На етапі **концептуального** проектування проект розглядається з погляду вимог користувачів і бізнесу. Закінчується концептуальне проектування визначенням набору сценаріїв використання системи. При **логічному** проектуванні проект розглядається з погляду розробників (проектної команди), рішення подається у вигляді набору сервісів. На етапі **фізичного** проектування

проект розглядається з погляду програмістів, уточнюються використовувані технології та інтерфейси.

Розробка – Developing. Створюється варіант вирішення проблеми у вигляді кодів і документації чергового прототипу, включаючи специфікації та сценарії тестування. Продукт готовий до зовнішнього тестування і стабілізації. Крім того, замовники, користувачі, співробітники служби підтримки і супроводу, а також ключові учасники проекту можуть заздалегідь оцінити продукт і вказати всі недоліки, які потрібно усунути до його постачання.

Стабілізація – Stabilizing. Підготовка до випуску остаточної версії продукту, доведення його до заданого рівня якості. Тут виконується комплекс робіт з тестування (виявлення й усунення дефектів), перевіряється сценарій розгортання продукту. Коли рішення стає досить стійким, проводиться його пілотна експлуатація в тестовому середовищі із залученням користувачів і застосуванням реальних сценаріїв роботи.

Розгортання – Deploying. Виконується встановлення рішення і необхідних компонентів оточення, проводиться його стабілізація в промислових умовах і передача проекту в руки групи супроводу. Крім того, аналізується проект в цілому на предмет рівня задоволеності замовника.

9.4.4. eXtreme Programming та інші гнучкі методології

eXtreme Programming. Методологія eXtreme Programming, розроблена Kent Beck, Ward Cunningham, Iron Jeffries сьогодні є найбільш відомою серед гнучких методологій. Екстремальне програмування як метод гнучкого проектування вперше було використано в ході роботи над проектом C3 (Chrysler Comprehensive Compensation System – система обліку виплат працівникам компанії Daimler Chrysler).

eXtreme Programming – технологія, що активно розвивається останнім часом, призначена для вирішення відносно невеликих завдань, відносно невеликими колективами професійних розробників в умовах жорсткого обмеження часу. Екстремальне програмування виникло як еволюційний метод розробки ПЗ "від низу до верху": від розробників і тестерів, виснажених важким процесом, документацією, метриками та іншим формалізмом. Вони не заперечували дисципліни, але не бажали марного дотримання формальних вимог і шукали нові швидкі та гнучкі підходи до розробки високоякісних програм. Іноді саме поняття гнучкої методології

ототожнюють із XP. Модель життєвого циклу XP є ітераційно-інкрементною моделлю швидкого створення (і модифікації) прототипів продукту, що задовольняють чергову вимогу (User Story).

До основних фаз моделі можна віднести (рис. 30):

"вкидання" архітектури – початковий етап проекту, на якому створюється бачення продукту, ухвалюються основні рішення з архітектури і використовуваних технологій. Результатом початкового етапу є **метафора (metaphor)** системи, яка в достатньо простому і зрозумілому команді вигляді повинна описувати основний механізм роботи системи;

історії використання (User Story) – етап збирання вимог, записуваних на спеціальних картках у вигляді сценаріїв виконання окремих функцій. Історії використання є вимогами для планування чергової версії і одночасної розробки **приймальних тестів (acceptance tests)** для її перевірки;

планування версії (релізу), яке проводиться на зборах за участю замовника шляхом вибору User Story, які увійдуть до наступної версії. Одночасно ухвалюються рішення, пов'язані з реалізацією версії. Мета планування – отримання оцінок того, що і як можна зробити за 1 – 3 тижні створення наступної версії продукту;

розробка версії (ітерації), яка проводиться відповідно до плану і включає тільки ті функції, які були відібрані на етапі планування;

приймальні тести, які проводяться за участю замовника, який бере участь у складанні тестів;

випуск релізу, згідно з яким розроблена невелика версія передається замовникові для використання або бета-тестування.

Після завершення циклу здійснюється перехід на наступну ітерацію розробки.

Крім того, в XP є декілька правил (технік), що характеризують особливості моделі його життєвого циклу і які повинні застосовувати всі члени команди:

живе планування (planning game). Його завдання – якнайшвидше визначити обсяг робіт, які потрібно зробити до наступної версії ПЗ. Рішення ухвалюється, по-перше, на основі пріоритетів замовника (тобто його потреб: що потрібно йому від системи для успішнішого ведення свого бізнесу) і, по-друге, на основі технічних оцінок (тобто оцінок трудомісткості розробки, сумісності з рештою елементів системи та ін.);

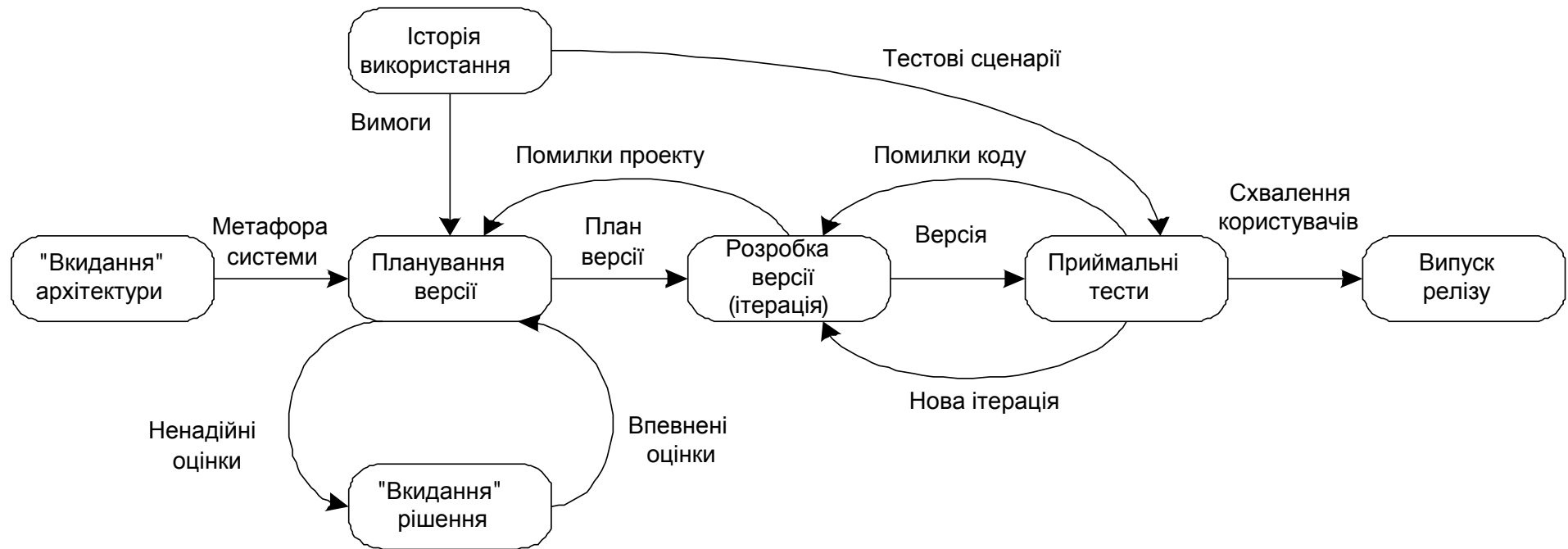


Рис. 30. Схеми моделі ЖЦ XP

часта зміна версій (small releases). Найперша дієва версія повинна з'явитися якнайшвидше і тут же повинна почати використовуватися. Наступні версії готуються через досить короткі проміжки часу (від декількох годин за незначних змін в невеликій програмі, до місяця – двох у випадку серйозної переробки великої системи);

метафора (metaphor) системи. Метафора в достатньо простому і зрозумілому команді вигляді повинна описувати основний механізм роботи системи. Це поняття нагадує архітектуру, але повинне набагато простіше, всього у вигляді однієї – двох фраз, описувати основну суть ухвалених технічних рішень;

прості проектні рішення (simple design). У кожен момент часу система повинна бути сконструйована настільки просто, наскільки це можливо. Не треба додавати функції заздалегідь – тільки після явного прохання про це. Уся зайва складність усувається, як тільки виявляється;

розробка на основі тестування (test-driven development). Розробники спочатку пишуть тести, потім намагаються реалізувати свої модулі так, щоб тести спрацьовували. Замовники заздалегідь пишуть тести, що демонструють основні можливості системи, щоб можна було побачити, що система дійсно запрацювала;

постійна переробка (refactoring). Програмісти постійно переробляють систему для усунення надмірної складності, збільшення зрозумілості коду, підвищення його гнучкості, але без змін в його поведінці, що перевіряється після кожної переробки тестів. При цьому перевага надається елегантнішим і гнучкішим рішенням, порівняно з тими, що просто дають потрібний результат. Невдало перероблені компоненти повинні виявлятися у ході виконання тестів і повертатися до останнього цілісного стану (разом із залежними від них компонентами);

програмування парами (pair programming). Кодування виконується двома програмістами на одному комп'ютері. Об'єднання в пари довільне і змінюється від завдання до завдання. Той, у чиїх руках клавіатура, намагається найкращим чином вирішити поточне завдання. Другий програміст аналізує роботу першого і дає поради, обмірковує наслідки тих чи інших рішень, нові тести, менш прямі, але гнучкіші рішення;

колективне володіння кодом (collective ownership). У будь-який момент будь-який член команди може змінити будь-яку частину коду. Ніхто не повинен виділяти свою власну сферу відповідальності, вся команда в цілому відповідає за весь код;

постійна інтеграція (continuous integration). Система збирається і проходить інтеграційне тестування якомога частіше, по кілька разів на день, щоразу, коли пара програмістів закінчує реалізацію чергової функції;

40-годинний робочий тиждень. Наднормова робота розглядається як ознака великих проблем у проекті. Не допускається наднормова робота 2 тижні підряд – це виснажує програмістів і робить їх роботу значно менш продуктивною;

включення замовника в команду (on-site customer). У складі команди розробників постійно знаходиться представник замовника, який доступний протягом усього робочого дня і здатний відповідати на всі питання про систему. Його обов'язком є достатньо оперативні відповіді на питання будь-якого типу, що стосуються функцій системи, її інтерфейсу, необхідної продуктивності, правильної роботи системи в складних ситуаціях, необхідності підтримувати зв'язок з іншими додатками та ін.;

використання коду як засобу комунікації. Код розглядається як найважливіший засіб спілкування всередині команди. Зрозумілість коду – один з основних пріоритетів. Дотримання стандартів кодування, що забезпечують таку зрозумілість, обов'язкове. Такі стандарти, крім зрозумілості коду, повинні забезпечувати мінімальність виразів (заборона на дублювання коду й інформації) і повинні бути прийняті всіма членами команди;

відкритий робочий простір (open workspace). Команда розміщується в одному, досить просторому приміщенні, для спрощення комунікації і можливості проведення колективних обговорень під час планування й ухвалення важливих технічних рішень;

зміна правил за потреби (just rules). Кожен член команди повинен прийняти перераховані правила, але в разі виникнення необхідності команда може змінити їх, якщо всі її члени прийшли до згоди з приводу цієї зміни.

Сфера застосування XP: використовувані в XP техніки розраховані на використання в рамках невеликих команд (не більше 10 програмістів) і, як наслідок, невеликих проектів. Більший розмір команди руйнує необхідну для успіху простоту комунікації і робить неможливим застосування багатьох перерахованих прийомів.

Перевагами XP є:

велика гнучкість, можливість швидко й акуратно вносити зміни до ПЗ у відповідь на зміни вимог і окремі побажання замовників;

висока якість отриманого в результаті коду і відсутність необхідності переконувати замовників у тому, що результат відповідає їх очікуванням.

Недоліками цього підходу є:

нездійсненність у такому стилі досить великих і складних проектів;
неможливість планувати терміни і трудомісткість проекту на досить довгу перспективу і чітко передбачити результати тривалого проекту в термінах співвідношення якості результату і витрат часу та ресурсів;

неприспосованість для тих випадків, у яких можливі рішення не знаходяться відразу на основі раніше отриманого досвіду, а вимагають проведення попередніх досліджень.

Прикладом зазначених переваг і недоліків XP може слугувати проект СЗ, який був розпочатий у січні 1995 року. Система створювалася мовою Smalltalk компанією, яка не впоралася із завданням. Якість програмного коду була настільки низькою, що Кент, якого запросила компанія, порадив почати все з початку. Проект почався з нуля тепер вже під його керівництвом і з березня 1996 року проходив з використанням цього методу. До сьогодні проект вийшов за межі бюджету і планів поетапної реалізації функцій. Команда розробників була скорочена. Протягом приблизно півроку після цього проект розвивався досить успішно. У серпні 1998 року з'явився прототип, який міг обслуговувати близько 10 000 службовців. Спочатку передбачалося, що проект завершиться в середині 1999 року і використовуватиметься для управління виплатами 87 000 службовцям компанії. Але він був зупинений в лютому 2000 року після 4-х років роботи з XP у зв'язку з повним недотриманням часових меж і бюджету. Створене ПЗ жодного разу не використовувалося для роботи з даними для більш ніж 10 000 службовців, хоча було показано, що воно впорається із даними 30 000 працівників компанії. Включений в команду проекту замовник звільнився через декілька місяців роботи за правилами методу XP, не витримавши навантаження. Проект так і не отримав адекватної заміни до свого кінця.

Crystal Clear. Crystal – сімейство методологій, що визначають необхідний ступінь формалізації процесу розробки залежно від кількості учасників і критичності завдань. Ці методології розроблені Алістером Кокберном (Alistair Cockburn) і називаються сімейством, оскільки автор переконаний, що різним проектам потрібні різні методології. Він увів наступну градацію проектів: на одній осі відкладається кількість зайнятих у проекті людей, на іншій – критичність помилок. Кожна з методологій "сімейства" призначена для певної клітинки сітки. Таким чином, проект, в якому зайнято 40 осіб і в якому компанія може дозволити собі втратити певну суму, працюватиме за іншою методологією, ніж проект для 6 розробників, від якого залежить існування компанії.

Методологія поступається XP за продуктивністю, зате максимально проста у використанні. Вимагає мінімальних зусиль для впровадження, оскільки орієнтована на людські звички. Вважається, що вона описує той природний порядок розробки ПЗ, який встановлюється в досить кваліфікованих колективах, якщо в них не займаються цілеспрямованим впровадженням іншої методології.

Основні **характеристики** методології:

ітеративна інкрементна розробка;

автоматичне регресійне тестування;

користувачі залучаються до активної участі в проекті;

склад документації визначається учасниками проекту;

як правило, використовуються засоби контролю версій коду.

Крім Crystal Clear в сімейство Crystal входить ще декілька методологій, призначених для виконання більших або критичніших проектів. Вони відрізняються дещо жорсткішими вимогами до обсягу документації і допоміжних процедур, такими як управління змінами і версіями.

Feature Driven Development. Методологія Feature Driven Development (функціонально-орієнтована розробка), яка має скорочену назву FDD, була розроблена Джефом Де Люка (Jeff De Luca) і визнаним авторитетом у галузі об'єктно-орієнтованих технологій Пітером Коадом (Peter Coad). Як і в решті адаптивних методологій, у ній робиться основний наголос на коротких ітераціях, кожна з яких слугує для опрацювання певної частини функціональності системи. Згідно з FDD, одна ітерація триває два тижні.

Насправді використовуване в FDD поняття функції або властивості системи (feature) достатньо близьке до поняття прецеденту використання, використовуваного в RUP. Чи не найістотніша відмінність – це додаткове обмеження: "кожна функція повинна допускати реалізацію не більш ніж за два тижні". Тобто якщо сценарій використання досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

FDD включає п'ять процесів:

розробку загальної моделі;

складання списку необхідних функцій системи;

планування роботи над кожною функцією;

проективання функції;

конструювання функції.

Перші три з них відносяться до початку проекту. Останні два – необхідно виконувати під час кожної ітерації. При цьому кожен процес розбивається на завдання і має критерії верифікації.

Розробники в FDD поділяються на "class owners" (власники класів) і "chief programmers" (старші програмісти). Старші програмісти залучають власників задіяних класів до роботи над черговою властивістю. Для порівняння, в XP немає персонально відповідальних за класи або методи. До роботи над будь-яким класом може залучатися будь-який із учасників розробки.

Робота над проектом передбачає часті збирання продукту і ділиться на ітерації, кожна з яких передбачає реалізацію певного набору функцій.

Dynamic System Development Method (DSDM) з'явився у Великобританії в 1994. Його заснував консорціум із 17 англійських компаній, які хотіли працювати з використанням RAD-технології і принципів ітеративної розробки. Зараз кількість його членів перевищує тисячу, причому багато хто з них перебуває за межами країни. Те, що DSDM розробляється цілим консорціумом, помітно відрізняє його від інших гнучких методологій. Ціла організація займається розробкою посібників з цієї методології, організацією навчальних курсів, програм акредитації тощо.

Проектування методологією DSDM починається з:

вивчення здійсненності програми для розуміння, чи підходить DSDM для даного проекту;

визначення сфери застосування програми, що відбувається на короткій серії семінарів, де програмісти дізнаються про сферу бізнесу, для якої буде розроблена програма. Тут же обговорюються основні положення, що стосуються архітектури майбутньої системи і плану проекту.

Далі процес ділиться на три взаємопов'язаних цикли:

цикл функціональної моделі відповідає за створення аналітичної документації та прототипів;

цикл проектування і конструювання – за приведення системи в робочий стан;

цикл реалізації забезпечує розгортання програмної системи.

Базові принципи, на яких будується DSDM, – це:

активна взаємодія з користувачами;
часті випуски версій;
самостійність розробників в ухваленні рішень;
тестування протягом усього циклу робіт.

Як і більшість інших гнучких методологій, DSDM використовує короткі ітерації тривалістю від двох до шести тижнів кожна. Особливий наголос робиться на високій якості роботи і адаптації до змін у вимогах.

9.5. Аналіз вимог до програмного забезпечення

У процесі розробки програмних систем головним завданням є аналіз вимог до програмного забезпечення. Правильно визначені вимоги на ранніх стадіях проектування є гарантією того, що система задовольнятиме вимоги зацікавлених у її розробці осіб.

Оскільки, звичайно, вимоги пов'язані одна з одною, то процес їх систематизації і подальшого виконання може віднімати значні сили. Причому завжди існує вірогідність, що якісь із вимог користувачів будуть не реалізовані або неправильно реалізовані. Для великих систем кількість цих вимог може бути дуже великою і при цьому вимоги можуть змінюватися.

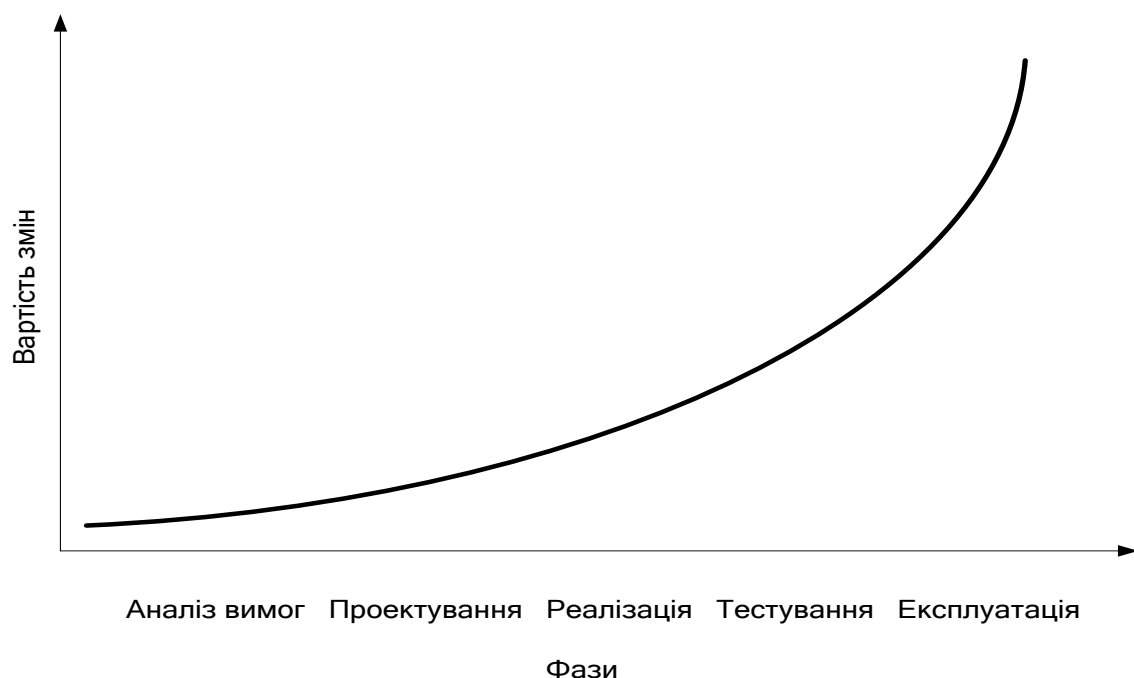


Рис. 31. **Графік залежності вартості внесення змін від фази ЖЦ**

Досвід розробки програмних проектів показує, що типовими причинами порушення термінів проектування і збільшення його бюджетів є наступні:

не всі вимоги замовників були виявлені;

вимоги не були чітко сформульовані;
не вдалося відстежити зміни вимог.

Усунення помилок у вимогах на стадії супроводу готового ПЗ коштує в декілька разів дорожче, ніж на стадії специфікації вимог (рис. 31). На виправлення помилок у вимогах, які виявляються на пізніх фазах проекту, витрачається 30 – 40% від загальної вартості бюджету проекту. Тому виявлення, аналіз та управління вимогами до програмного проекту є найважливішими завданнями проектної діяльності.

Програмна вимога (Software Requirement) – це можливість, яку будь-хто очікує від даного ПЗ. У стандарті IEEE Standard Glossary of Software Engineering Terminology, 1990 вимога визначається таким чином:

1. Умова або можливість, необхідна користувачеві для вирішення завдань або досягнення мети.

2. Умова або можливість, яку повинна мати система або її компонент, відповідні договору, стандарту, специфікації або іншому офіційному документу.

3. Документоване подання умови або можливості, перераховані в попередніх пунктах.

Реалізація кожної окремої вимоги є додаванням в систему певної здатності, якій та повинна відповідати. Процес управління вимогами охоплює декілька діяльностей: виявлення, організацію і документування. Існують перевірені рішення, які дозволяють гарантувати ефективніше ведення проекту, засноване на грамотному управлінні вимогами. Вони спираються на наступні аспекти:

організований підхід до управління вимогами;

взаємодію учасників проекту на базі виявлених і затверджених вимог;
ранжування вимог за пріоритетом, фільтрація їх за необхідними параметрами і виявлення залежності між ними для контролю за змінами;

об'єктивну оцінку реалізованого функціонала і отриманої продуктивності;

ранній прогноз різних невідповідностей і розбіжностей;

використання інструментальних засобів для організації ефективного процесу управління вимогами.

Вимоги, що висувуються до ПЗ, можна класифікувати наступним чином:

усвідомлені (відомі, висловлені) та неусвідомлені (забуті, невисловлені);

функціональні (властивості продукту), які визначають, що повинен виконувати програмний продукт, і нефункціональні, такі, що мають відношення до якості самого ПЗ (до зовнішнього інтерфейсу, до продуктивності, до якості ПЗ, пов'язані з розробкою проекту);

замовника (С-вимоги) і розробника (D-вимоги);

бізнес-вимоги (вищий рівень), вимоги користувача (середній рівень), вимоги розробника (нижчий рівень).

Можна виділити три рівні вимог (рис. 32).

На верхньому рівні знаходяться **бізнес-вимоги (business requirements)**. Бізнес-вимоги відбивають високорівневі цілі організації. Їх формулюють топ-менеджери. Ці вимоги формулюються в документах "Бачення проекту" (Vision), "Глосарій проекту" (Glossary) та "Бізнес-правила" (Business Rules).

На середньому рівні знаходяться **вимоги користувачів (user requirements)**. Вони описують цілі і завдання, які дозволить вирішити система користувачам. Ці вимоги описуються за допомогою документів: "Варіанти використання" (Use Case) та "Додаткові специфікації" (Supplementary Specifications).

На нижньому рівні знаходяться системні вимоги, які формулюються за допомогою функціональних вимог і нефункціональних вимог і містяться в документі Специфікації вимог до ПЗ (SRS).

Складність аналізу вимог обумовлена наступними основними причинами, такими як:

велика кількість осіб, зацікавлених у розроблюваному ПЗ, вимоги яких потрібно виявити і зафіксувати;

різноманітність типів вимог, кожен із яких вимагає специфічного опису, своїх атрибутів і ступеня деталізації;

необхідність створення й підтримки складної ієрархічної структури вимог;

необхідність трасування вимог, тобто виявлення взаємозв'язку між вимогами різних типів;

зміна вимог у ході виконання проекту. Основним засобом документування вимог є текст природною мовою. Тому при документуванні вимог виникають проблеми, пов'язані з неоднозначною інтерпретацією і слабкою структуризацією інформації.

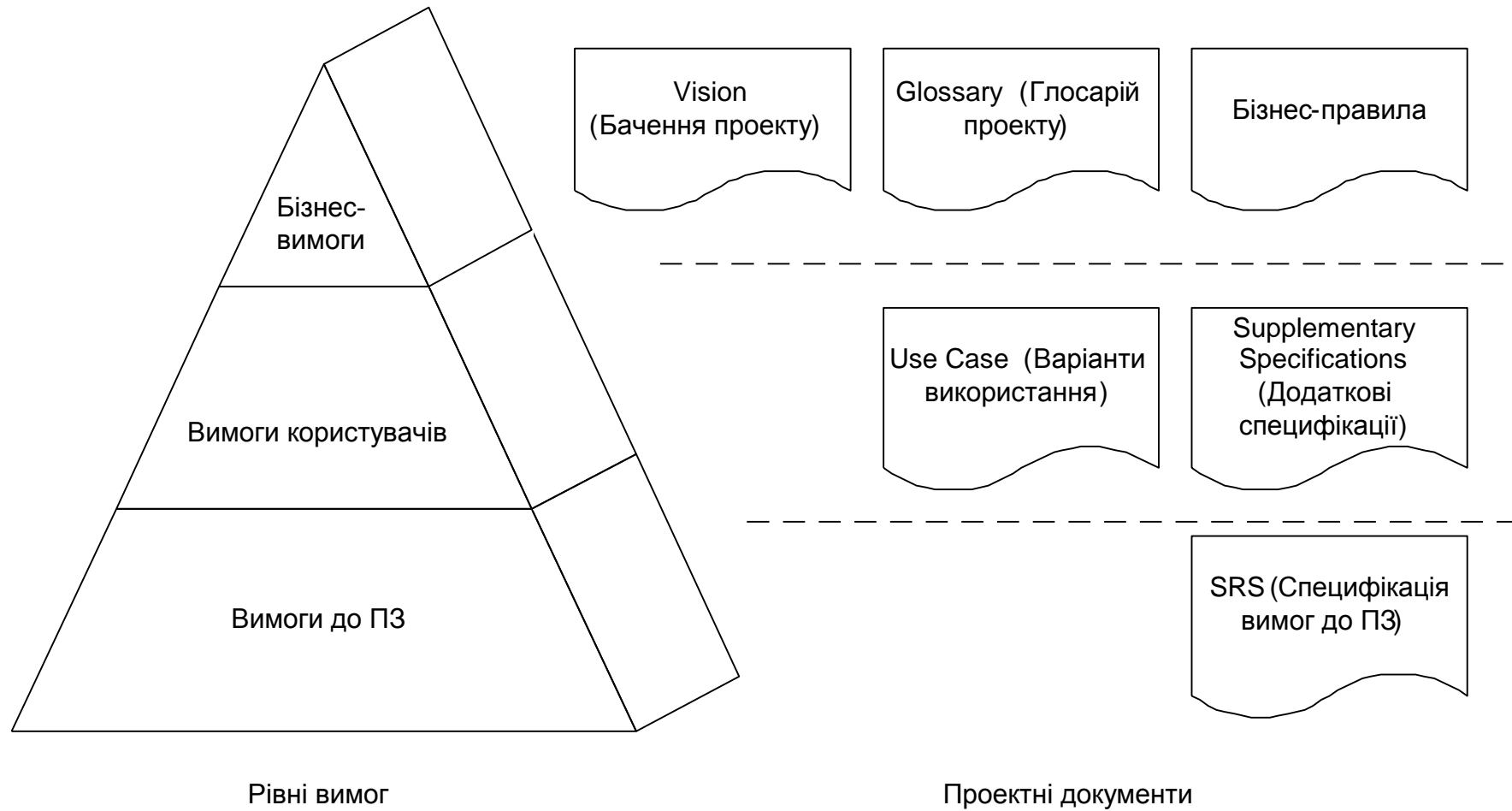


Рис. 32. Рівні вимог

Вимоги, що документуються, повинні мати наступні властивості:
бути чітко вираженими;
бути доступними;
бути пронумерованими;
супроводжуватися підтверджувальними тестами;
передбачатися проектом;
бути врахованими програмним кодом;
бути протестованими окремо;
бути протестованими разом з іншими вимогами;
бути опротестованими після збирання додатка.

Процес створення вимог згідно із SWEBOOK включає наступні складові:
Requirements Elicitation (встановлення вимог);
Requirements Analysis(аналіз вимог);
Requirements Specification (специфікація вимог);
Requirements Validation (перевірка вимог).

Для роботи з вимогами і документами, в яких вони відбиваються, відстеження їх змін фірмою IBM був розроблений програмний продукт Rational RequisitePro.

У процесі документування вимог в RequisitePro розглядаються наступні їх характеристики:

типи вимог (**Requirements Types**): бізнес-вимоги, функціональні, нефункціональні вимоги і т. д.;

види типів вимог (ієрархічні: батько – нащадок);

атрибути (**Attributes**) типів вимог і їх значення (**Values per Attributes**).

За замовчуванням у **RequisitePro** для вимог встановлюються наступні атрибути: пріоритет (**Priority**), стан (**Status**), вартість (**Cost**), складність при реалізації (**Difficulty**), стабільність (**Stability**), призначений (**Assigned to**), джерело (**Origin**);

залежності між вимогами (**trace to, trace from**).

На рис. 33 наведена структура програмного продукту **RequisitePro**. Інформація про всі вимоги зберігається в БД незалежно від того, в якому вигляді вона відображається: перегляди (**View**) **RequisitePro**, документи **RequisitePro** або **MS Word Document**.

У методикі управління вимогами в RequisitePro в процесі розробки програмних систем входить:

планування проекту;

розробка корпоративного стандарту;

реалізація проекту.

Планування проекту включає виконання наступних робіт:

- вибір методики розробки ПЗ, визначення документів і вимог тощо;
- вибір призначеного для користувача середовища (автономного, багатокористувальницького режиму);
- вибір СУБД (Access, SQL Server, Oracle);
- визначення способу створення вимог (лише у БД, лише в документах, в БД і документах);
- визначення місця розташування проекту;
- розробку плану проекту управління вимогами, в якому наводиться: інформація про розробників проекту (опис і закріплення ролей за виконавцями), артефакти вимог (типи документів, типи вимог, їх атрибути і значення атрибутів, трасування вимог), порядок управління змінами вимог.

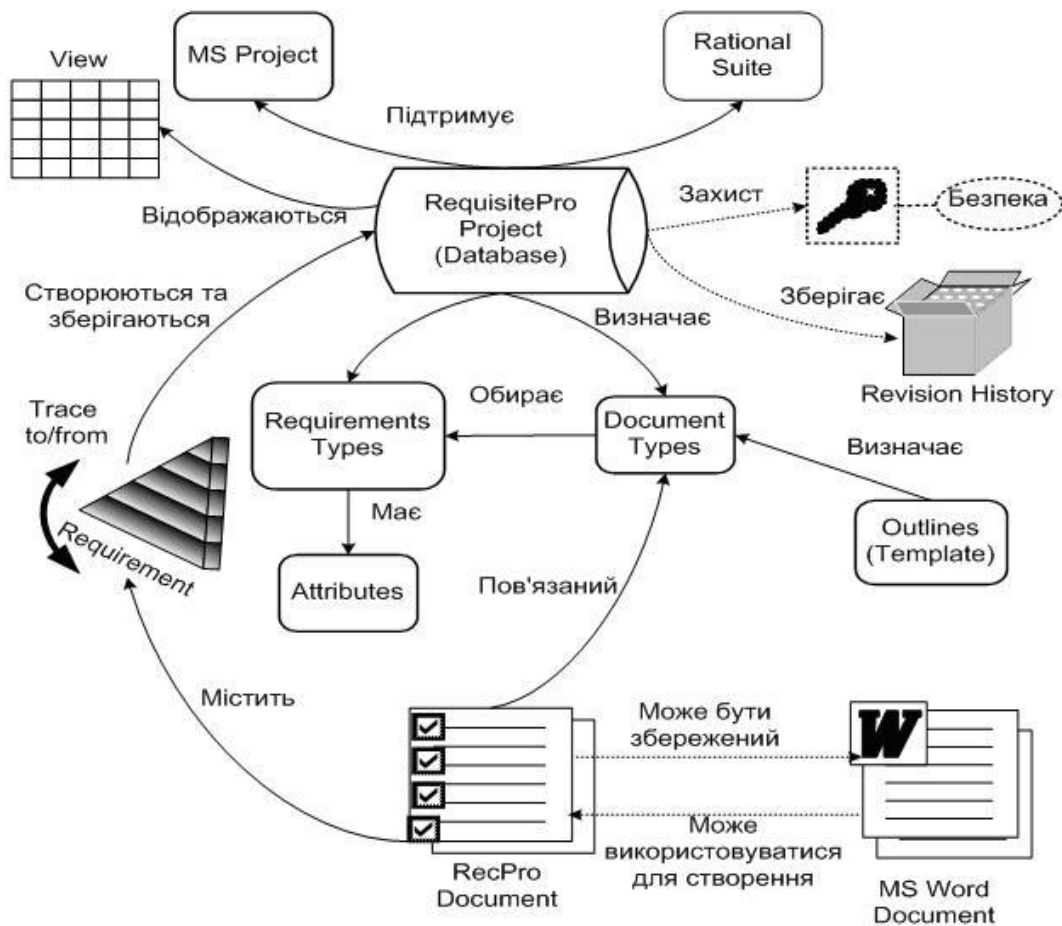


Рис. 33. Структура програмного продукту Rational RequisitePro

У розробку корпоративного стандарту роботи в середовищі IBM Rational RequisitePro входить:

вибір життєвого циклу програмних засобів;
визначення складу документів, підтримуваних в IBM Rational RequisitePro;
розробка шаблонів (**template**) документів з подальшим їх розміщенням у папці шаблонів **Outline**;
визначення типів вимог їх атрибутів і значень для шаблонів документів;
задавання матриць трасувань.

Реалізація проекту в IBM Rational RequisitePro передбачає:

створення проекту;
створення шаблонів документів;
задавання типів вимог;
задавання атрибутів типів вимог;
задавання типів документів;
створення документів;
створення вимог у документах і (або) в БД і їх атрибутів;
створення переглядів вимог, сортування вимог, фільтрація вимог,
задавання запитів до БД вимог, метрики;
задавання зв'язків між вимогами, підозрілих зв'язків, різних можливостей під час перегляду зв'язків;
стеження за змінами вимог;
створення списку змін вимог;
забезпечення безпеки проекту;
розробка специфікації вимог до ПЗ.

Вимоги до програмного забезпечення оформляються відповідно до стандартів, що діють:

у вигляді документа ТЗ – "Технічне завдання", на основі вітчизняного стандарту ГОСТ 34.602 – 90;

у вигляді документа SRS – "Специфікація вимог до системи", на основі стандарту IEEE 830 –1993.

Який із типів документів ТЗ або SRS використовувати для документування вимог, залежить від корпоративного стандарту і від вимог замовника.

У процесі створення ПЗ RequisitePro взаємодіє з іншими інструментами RUP, як це показано на рис. 34.

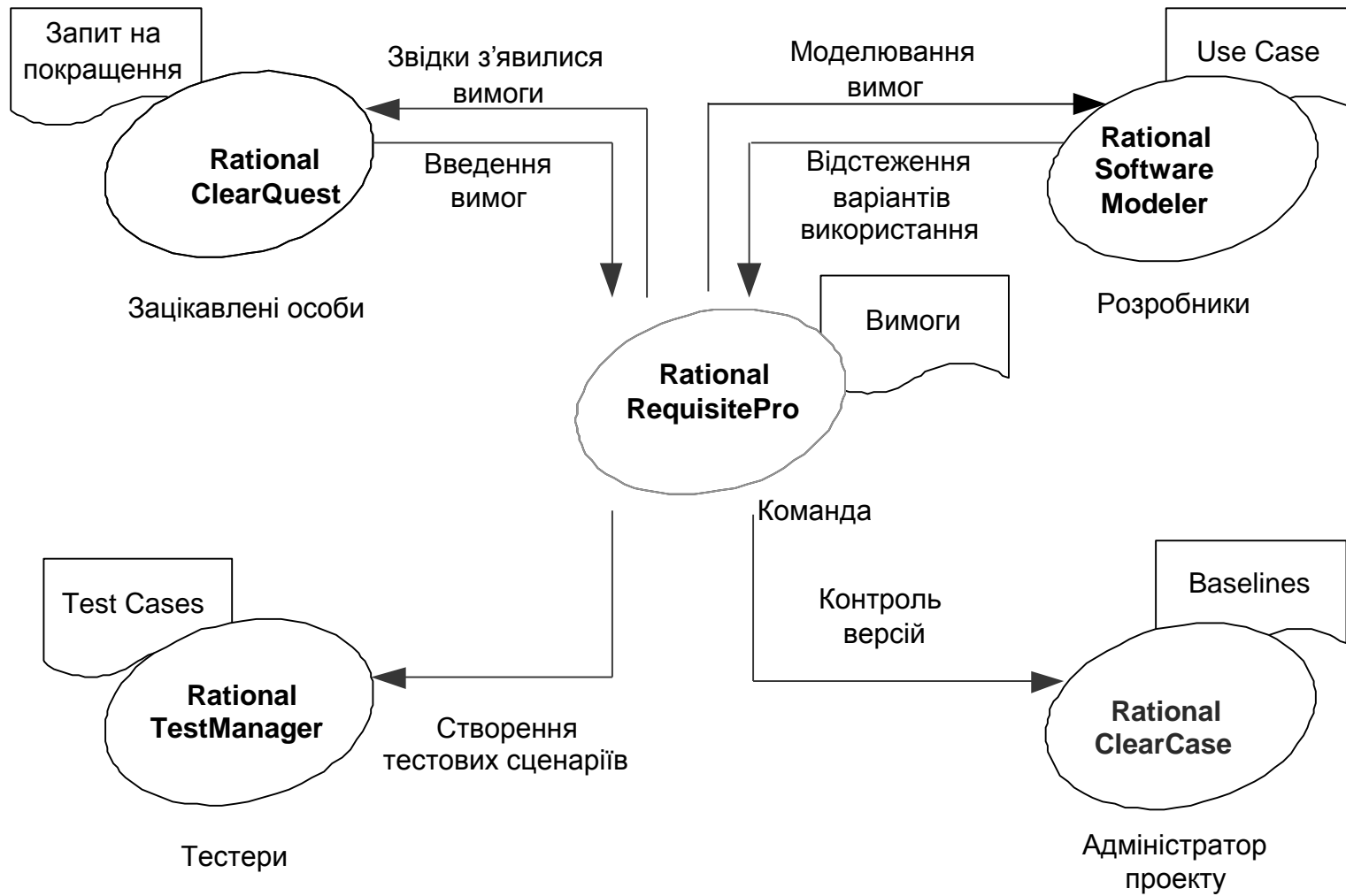


Рис. 34. Зв'язок Rational RequisitePro з іншими інструментами RUP

Вимоги повинні бути інтегровані з іншими процесами життєвого циклу розробки ПЗ, що забезпечує чіткі комунікації в проекті. Це економить ресурси за рахунок мінімізації змін уже готових матеріалів. Rational RequisitePro дозволяє використовувати вимоги (зі свого репозиторію) в інших інструментах Rational Software.

Вимоги є вхідною інформацією для проектування ПЗ. Інтеграція RequisitePro з Rational Software Modeler пов'язує вимоги з моделями варіантів використання в Modeler. Розробники і проектувальники можуть безпосередньо мати доступ до специфікацій варіантів використання, атрибутів RequisitePro безпосередньо зі звичного середовища Modeler.

У процесі розробки неминуча поява нових вимог. Інтеграція Rational RequisitePro і Rational ClearQuest полегшує внесення змін до проекту, асоціюючи запит на зміну з новою вимогою, яка його задовольняє. Доступ до інформації про вимоги можливий як із RequisitePro, так і з ClearQuest, включаючи, наприклад, такі атрибути, як пріоритет, статус, трудомісткість.

Інтеграція з Rational ClearCase дозволяє контролювати версії вимог нарівні з іншими артефактами проекту і створювати базові версії вимог.

Інтеграція з Rational TestManager гарантує, що вимоги використовуються як вхідна інформація для створення сценарію тестування тестером і фахівцем з контролю якості.

Контрольні запитання

1. Охарактеризуйте стани життєвого циклу продукту.
2. Дайте визначення програмної інженерії.
3. Що таке життєвий цикл програмного продукту?
4. Які основні риси ЖЦ ПЗ?
5. Що таке стандарт?
6. Що таке заява постачальника про відповідність, сертифікація відповідності?
7. Які види стандартів існують? Наведіть приклади кожного з видів стандартів.
8. Охарактеризуйте основні організації-розробники стандартів в галузі програмної інженерії.
9. Перерахуйте найбільш відомі стандарти в галузі програмної інженерії.
10. Що таке процес, дія, завдання?
11. Які типи процесів і конкретні процеси виділяють в ЖЦ ПЗ?
12. Що таке верифікація? З яких дій і завдань складається цей процес?
13. Що таке валідація? З яких дій і завдань складається цей процес?
14. Що таке модель життєвого циклу ПЗ?
15. Що таке фаза проекту, чим вона відрізняється від процесу?
16. Які існують типи традиційних моделей ЖЦ ПЗ? У чому суть цих моделей, які їх переваги, недоліки, область застосування?

17. У чому основна відмінність "живих" технологій проектування ПЗ від "важких"?

18. Які основні характеристики, підхід до розробки та принципи методології RUP?

19. Охарактеризуйте модель ЖЦ RUP. Що таке динамічна і статична структури моделі ЖЦ RUP?

20. Охарактеризуйте основні концепції і принципи методології MSF.

21. Які фази і віхи включає модель ЖЦ MSF?

22. Які особливості, переваги і недоліки екстремального програмування? Де його застосовують?

23. Охарактеризуйте модель ЖЦ XP.

24. Що таке програмна вимога? Як вимога визначається в стандарті IEEE?

25. Наведіть рівні вимог і відповідні їм проектні документи.

26. Чим обумовлена складність аналізу вимог?

27. Назвіть складові процесу створення вимог згідно SWEBOOK?

28. Яка структура програмного продукту RequisitePro?

29. Які характеристики вимог підтримуються RequisitePro?

30. Яка послідовність дій управління вимогами під час планування проекту в IBM Rational RequisitePro?

31. Яка послідовність дій управління вимогами в процесі розроблення корпоративного стандарту в IBM Rational RequisitePro?

32. У яких документах відбиваються вимоги до проектного ПЗ?

33. З якими інструментами RUP взаємодіє RequisitePro під час проектування ПЗ?

10. Канонічне проектування ІС

10.1. Технологія передпроектного обстеження

10.1.1. Склад стадій і етапів канонічного проектування ІС

Канонічне проектування ІС відбиває особливості ручної технології індивідуального (оригінального) проектування, здійснюваного на рівні виконавців без використання яких-небудь інструментальних засобів, які дозволяють інтегрувати виконання елементарних операцій.

Процес каскадного проектування в життєвому циклі ІС відповідно до використовуюваного в нашій країні ГОСТ 34.601-90 "Автоматизовані системи. Стадії створення" включає наступні стадії та етапи проектування (табл. 6).

Допускається виключати стадію "Ескізний проект" і окремі етапи робіт на всіх стадіях, об'єднувати стадії "Технічний проект" і "Робоча документація" в одну стадію "Техноробоче проектування". Залежно від специфіки створюваних ІС і умов їх створення дозволяється виконувати окремі етапи робіт до завершення попередніх стадій, паралельне в часі виконання етапів робіт, включення нових етапів робіт.

З метою вивчення взаємозалежних прийомів і методів канонічного проектування ІС перераховані стадії можна згрупувати в часто використовувані на практиці чотири стадії процесу розробки ІС (рис. 35).

Таблиця 6

Стадії та етапи створення ІС

Стадії	Етапи робіт
1. Формування вимог до ІС	1.1. Обстеження об'єкта і обґрунтування необхідності створення ІС. 1.2. Формування вимог користувача до ІС. 1.3. Оформлення звіту про виконану роботу і заявки на розробку ІС (тактико-технічного завдання)
2. Розроблення концепції ІС	2.1. Вивчення об'єкта. 2.2. Проведення необхідних науково-дослідних робіт. 2.3. Розроблення варіантів концепції ІС і вибір варіанта концепції ІС, що задовольняє вимоги користувача. 2.4. Оформлення звіту про виконану роботу
3. Технічне завдання	3.1. Розроблення і затвердження технічного завдання на створення ІС
4. Ескізний проект	4.1. Розроблення попередніх проектних рішень для системи та її частин. 4.2. Розроблення документації на ІС та її частини
5. Технічний проект	5.1. Розроблення проектних рішень для системи та її частин. 5.2. Розроблення документації на систему та її частини. 5.3. Розроблення і оформлення документації на постачання виробів для комплектування ІС і (або) технічних вимог (технічних завдань) на їх розробку. 5.4. Розроблення завдань на проектування в суміжних частинах проекту об'єкта автоматизації
6. Робоча документація	6.1. Розроблення робочої документації на систему та її частини. 6.2. Розроблення або адаптація програм
7. Введення в дію	7.1. Підготовка об'єкта автоматизації до введення ІС в дію. 7.2. Підготовка персоналу. 7.3. Комплектація ІС виробами, що постачаються (програмними і технічними засобами, програмно-технічними комплексами, інформаційними виробами). 7.4. Будівельно-монтажні роботи. 7.5. Пусконаладжувальні роботи. 7.6. Проведення попередніх випробувань. 7.7. Проведення дослідної експлуатації. 7.8. Проведення приймальних випробувань
8. Супровід ІС	8.1. Виконання робіт відповідно до гарантійних зобов'язань. 8.2. Післягарантійне обслуговування

Традиційно етапи дослідження предметної сфери – підприємства, організації, обґрунтування проекту ІС для нього і розробки технічного завдання – поєднують терміном "Передпроектна стадія" ("Передпроектне обстеження"), оскільки результати виконання робіт на даних етапах не є закінченим проектним рішенням. Основне призначення "Передпроектної стадії" полягає в обґрунтуванні економічної доцільності створення ІС і формулюванні вимог до неї.

На **передпроектній стадії** прийнято виділяти два основних етапи: збору матеріалів обстеження; аналізу матеріалів обстеження, розробки техніко-економічного обґрунтування (ТЕО) і технічного завдання (ТЗ).

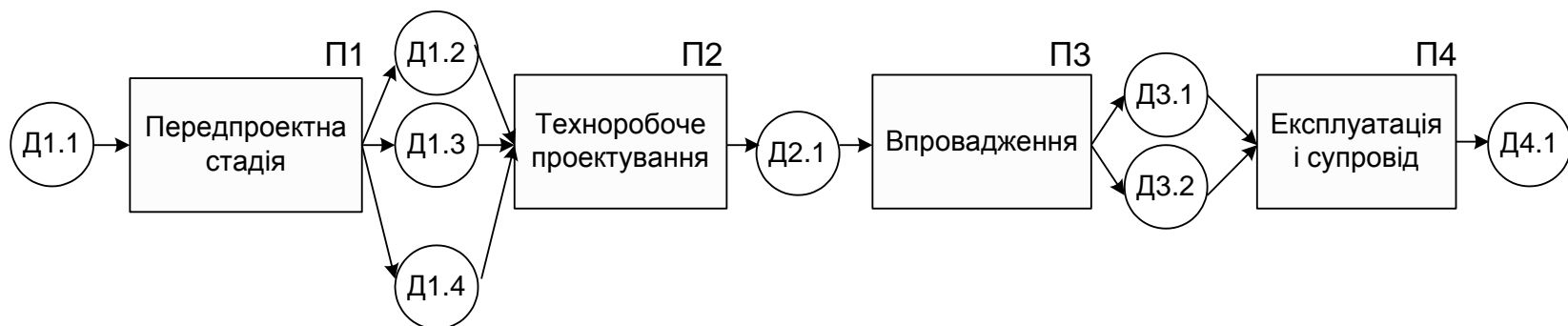


Рис. 35. Стадії канонічного проектування ІС

Умовні позначення:

Д1.1 – предметна сфера;

Д1.2 – матеріали обстеження;

Д1.3 – ТЕО, ТЗ на проектування;

Д1.4 – ескізний проект;

Д2.1 – техноробочий проект (ТРП);

Д3.1 – виправлений ТРП, переданий в експлуатацію;

Д3.2 – акт приймання проекту в промислову експлуатацію;

Д4.1 – модернізований ТРП.

У результаті виконання першого етапу **"Збір матеріалів обстеження"** проєктувальники одержують матеріали обстеження (Д1.2), що повинні містити повну й достовірну інформацію, яка описує досліджувану предметну область – підприємство, у тому числі: мету функціонування; організаційну структуру системи й об'єкта управління, тобто його управлінські відділи, цехи, склади й господарські служби; функції управління, виконувані в цих підрозділах, технологічні процеси обробки управлінської та економічної інформації, що відбуваються в них, а також матеріальні потоки і процеси їхньої обробки, ресурсні обмеження.

На основі цих матеріалів розробляються два документи: "Техніко-економічне обґрунтування проєктних рішень", що містить розрахунки й обґрунтування необхідності розробки ІС для підприємства й обраних технологічних і проєктних рішень (Д1.3), і "Технічне завдання", до складу якого входять вимоги до створюваної системи і її окремих компонентів: програмного, технічного й інформаційного забезпечення і цільової настанови на проєктування нової системи (Д1.4).

Для складних ІС іноді на цій стадії включають третій етап – **розробку "Ескізного проєкту"**. На етапі "Ескізного проєкту" сформульовані раніше вимоги є основою для розробки попередніх рішень для ІС у цілому й окремих видів забезпечення. Ці рішення проробляються на логічному рівні, включаючи алгоритми обробки інформації, опис інформаційних потреб користувачів на рівні назв документів і показників.

Друга стадія **"Техноробоче проєктування"** виконується в два етапи: технічне проєктування; робоче проєктування.

На етапі **технічного проєктування** виконуються роботи з логічної розробки і вибору найкращих варіантів проєктних рішень, у результаті чого створюється "Технічний проєкт". Етап **робочого проєктування** пов'язаний із фізичною реалізацією обраного варіанта проєкту й одержанням документації "Робочого проєкту". За наявності досвіду проєктування ці етапи іноді поєднуються в один, у результаті виконання якого одержують "Техноробочий проєкт" (ТРП) – Д2.1.

Третя стадія **"Впровадження проєкту"** містить у собі три етапи: підготовки об'єкта до впровадження проєкту; дослідного впровадження проєкту; здачі проєкту в промислову експлуатацію.

На етапі **"Підготовка об'єкта до впровадження проєкту"** здійснюється комплекс робіт з підготовки підприємства до впровадження розробленого проєкту ІС. На етапі **"Дослідне впровадження"**

здійснюють перевірку правильності роботи деяких частин проекту й одержують виправлену проектну документацію і "Акт про проведення дослідного впровадження". На етапі **"Здача проекту в промислову експлуатацію"** здійснюють комплексну системну перевірку всіх частин проекту, у результаті якої одержують дороблений "Техноробочий проект" (ДЗ.1) і "Акт приймання проекту в промислову експлуатацію" (ДЗ.2).

Четверта стадія – **"Експлуатація і супровід проекту"** включає етапи: експлуатації проекту; супроводу й модернізації проекту.

На етапі **"Експлуатація проекту"** одержують інформацію про роботу всієї системи в цілому й окремих її компонентів і збирають статистичні дані про збої системи у вигляді рекламацій і зауважень, які накопичуються для виконання наступного етапу.

На етапі **"Супровід проекту"** виконуються два види робіт: ліквідуються наслідки збоїв у роботі системи і виправляються помилки, не виявлені під час впровадження проекту, а також здійснюється модернізація проекту. У процесі модернізації проект або допрацьовується, тобто розширюється за складом підсистем і завдань, або виконується перенесення системи на іншу програмну або технічну платформу з метою адаптації її до зовнішніх і внутрішніх умов функціонування, що змінюються, у результаті чого одержують документи модернізованого "Техноробочого проекту" (Д4.1).

Вивчаючи наявну економічну систему, розробники повинні уточнити межі вивчення системи, визначити коло користувачів майбутньої ІС різних рівнів і виділити класи та типи об'єктів, що підлягають обстеженню і наступній автоматизації. Далі будуть розглянуті питання стосовно складу і змісту робіт на передпроектній стадії створення ІС.

10.1.2. Об'єкти обстеження в процесі створення ІС

Найважливішими **об'єктами обстеження** можуть бути:

структурно-організаційні елементи підприємства (наприклад, відділи, управління, цехи, ділянки, робочі місця);

функціональна структура, господарські процеси і процедури;

стадії (технічна підготовка, постачання, виробництво, збут) і елементи господарського процесу (засоби праці, предмети праці, ресурси, продукція, фінанси);

компоненти потоків інформації (документи, показники, файли, повідомлення). У випадку канонічного проектування основною одиницею обробки даних є **завдання**. Тому функціональна структура проблемної сфери на стадії передпроектного обстеження вивчається в розрізі

розв'язуваних завдань і комплексів завдань. При цьому завдання в змістовому аспекті розглядається як сукупність операцій перетворення певного набору вихідних даних для одержання результатної інформації, необхідної для виконання функції управління або ухвалення управлінського рішення. У більшості випадків вихідні дані й результати їхніх перетворень подаються у формі економічних документів; технології, методи і технічні засоби перетворення інформації; матеріальні потоки і процеси їх обробки.

10.1.3. Склад і зміст робіт на етапі "Збір матеріалів обстеження"

Основною **метою** виконання першого етапу передпроектного обстеження є:

виявлення основних параметрів предметної сфери (наприклад, підприємства чи його частини);

встановлення умов, у яких буде функціонувати проект ІС;

виявлення вартісних і часових обмежень на процес проектування.

На цьому етапі проектувальниками виконується ряд технологічних операцій:

попереднє вивчення предметної сфери;

вибір технології проектування;

вибір методу проведення обстеження;

вибір методу збору матеріалів обстеження;

розробка програми обстеження;

розробка плану-графіка збору матеріалів обстеження;

збір і формалізація матеріалів обстеження.

Виконання роботи "Попереднє вивчення предметної сфери" має своєю метою на основі загальних відомостей про об'єкт виявити попередні розміри обсягів робіт із проектування, склад вартісних і часових обмежень на процеси проектування, а також знайти приклади розробок проектів ІС для аналогічних систем.

Важливою роботою, яка визначає всі наступні роботи з обстеження об'єкта і проектування ІС, є "Вибір технології проектування". На сьогодні існує кілька типів технологій проектування: технологія оригінального, типового, автоматизованого і змішаного варіантів проектування.

Основними обмеженнями при виборі технології можуть слугувати: наявність коштів на придбання і підтримку обраної технології, обмеження за часом проектування, наявність відповідних інструментальних засобів і можливість забезпечення підтримки їхньої експлуатації власними силами, наявність фахівців відповідної кваліфікації. Результатом виконання цієї роботи є одержання опису обраної технології, методів і засобів проектування.

10.1.4. Методи проведення обстеження

Перед початком робіт із проведення обстеження необхідно вибрати метод проведення обстеження. Усі методи можна об'єднати в групи за наступними ознаками (рис. 36):

за **метою обстеження** виділяють метод організації локального проведення обстеження, використовуваний для розробки проекту окремого завдання або для комплексу завдань, і метод системного обстеження об'єкта, застосовуваний для вивчення всього об'єкта з метою розробки для нього проекту ІС у цілому;

за **кількістю виконавців**, що проводять обстеження, застосовується індивідуальне обстеження, здійснюване одним проектувальником, і бригадне з виділенням ряду бригад – виконавців, які вивчають усі підрозділи підприємства, і однієї координаційної бригади;

за **ступенем охоплення предметної сфери** застосовують метод суцільного обстеження, що охоплює всі підрозділи економічної системи, і вибіркове, застосовуване за наявності типових за структурою підрозділів (наприклад, цехів чи складів);

за **ступенем одночасності виконання робіт** першого і другого етапів передпроектної стадії (стосовно етапів) виділяють метод послідовного проведення робіт, за якого проектувальники спочатку збирають дані про предметну сферу, а потім їх вивчають (часто застосовують за відсутності досвіду у виконанні такого виду робіт), і метод паралельного виконання робіт, коли одночасно зі збором відбувається вивчення отриманих матеріалів обстеження, що значно скорочує час на проведення робіт на передпроектній стадії і підвищує якість одержуваних результатів.

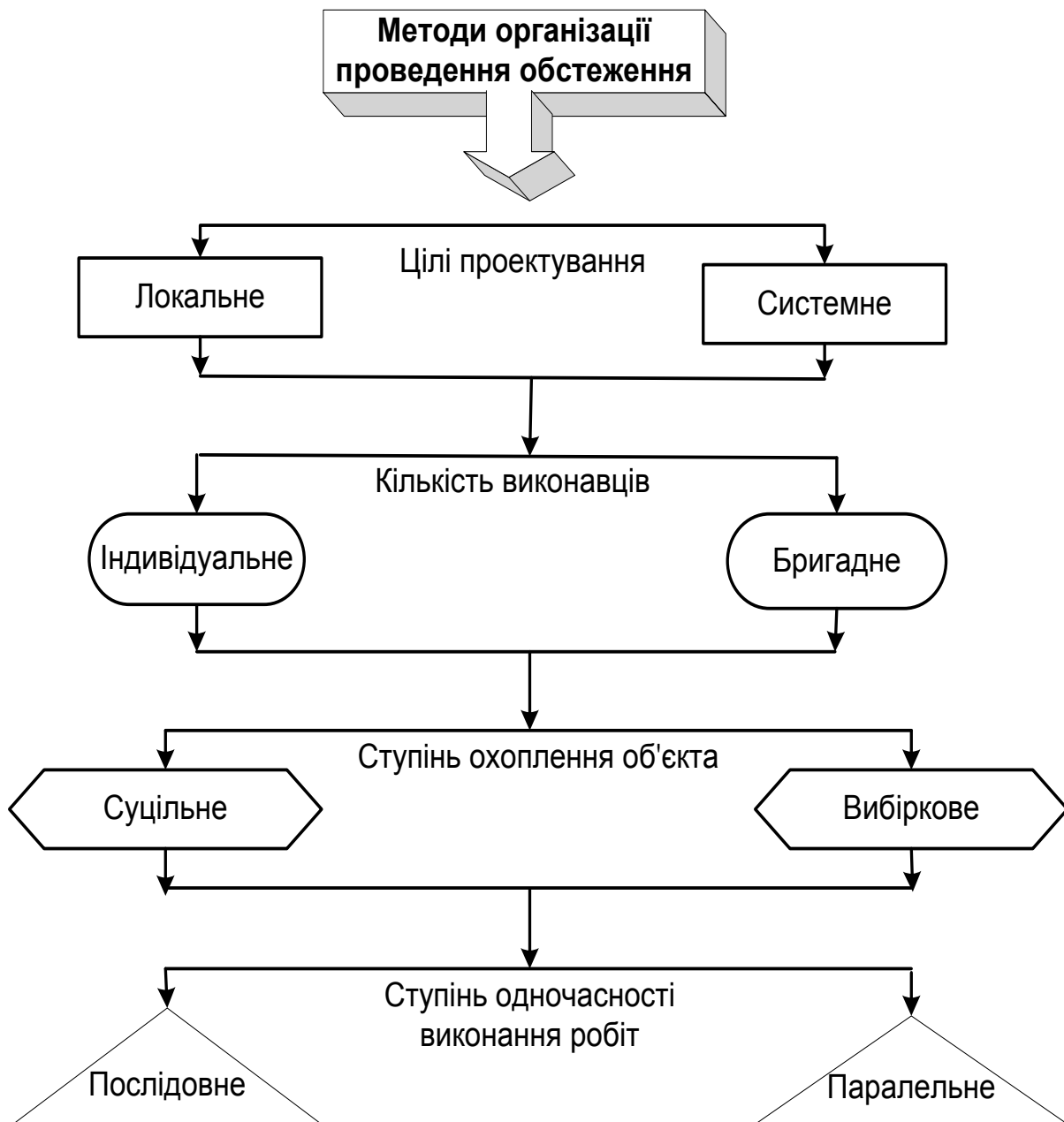


Рис. 36. Схеми класифікації методів проведення обстеження

Виконання робіт з обстеження предметної області в якому-небудь підрозділі і збору матеріалів можна здійснювати на основі попереднього проведення вибору методів збору матеріалів обстеження, які можна розподілити на дві групи (рис. 37):

виконувати силами **фахівців предметної області**, яким пропонується або заповнювати зошит-щоденник на виконуваних ними роботах, або провести документований опис робочого місця, або використовувати метод самофотографії робочого дня, що дозволяє виявити склад операцій і одержувати при цьому документи;

виконувані силами **проектувальників-виконавців**, які включають методи проведення бесід і опитувань, аналізу матеріалів обстеження, особистих спостережень, фотографії робочого дня і хронометражу робочого часу фахівця в процесі виконання ним тієї чи іншої роботи.

Метод документальної інвентаризації управлінських робіт полягає в тому, що на кожну роботу окремо відкривається спеціальна карта обстеження, в якій наводяться всі основні дані про роботу, що реєструється, чи документи, що складаються.

Метод самофотографії робочого дня полягає в тому, що спостереження має більш детальний характер і відбувається в короткий термін. Цей метод дає відомості про найбільш трудомісткі чи типові окремі роботи, які використовуються для визначення загальної трудомісткості виконання всіх робіт.

Метод ведення індивідуальних зошитів-щоденників. Записи в щоденнику здійснюються виконавцем протягом місяця щодня відразу ж після виконання чергової роботи.

Метод аналізу операцій полягає в розчленуванні розглянутого ділового процесу, роботи на їх складові частини, завдання, розрахунки, операції і навіть їх елементи. Після цього аналізується кожна частина окремо, виявляються повторюваність окремих операцій, багаторазове звертання до однієї й тієї ж операції, ступінь їх залежності одна від одної.

Метод особистого спостереження використовується, якщо досліджуване питання зрозуміле і необхідне лише уточнення деталей без істотного відриву виконавців від роботи.

Метод опитування виконавців на робочих місцях використовується в процесі збору відомостей безпосередньо у фахівців шляхом бесід, що вимагають ретельної підготовки. Заздалегідь складають список співробітників, з якими мають намір розмовляти, розробляють перелік питань про роль і призначення робіт у діяльності об'єкта, порядок їх виконання.

Метод аналізу наданого матеріалу застосовується в основному під час з'ясування таких питань, на які не можна одержати відповідь від виконавців.

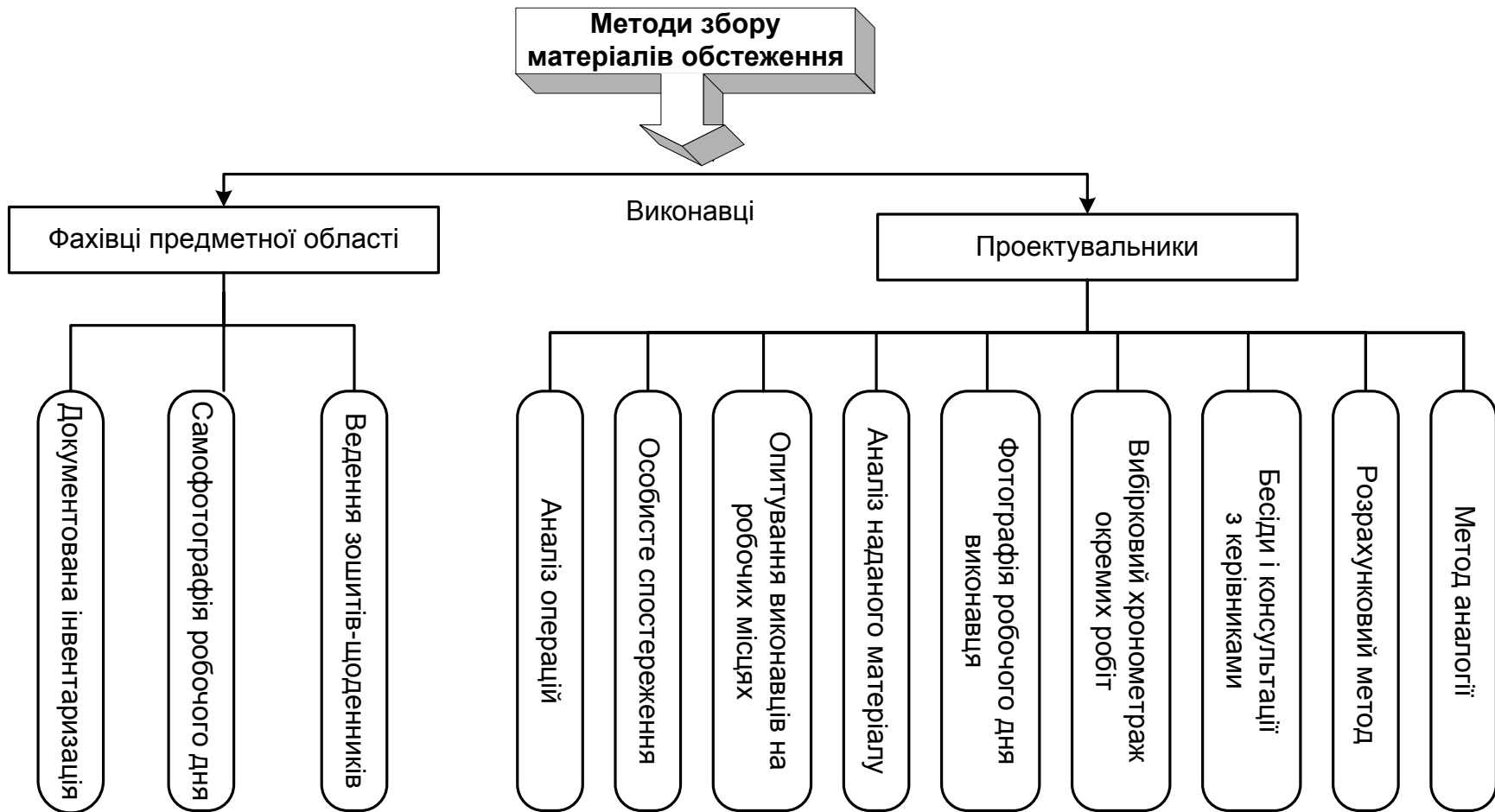


Рис. 37. Класифікація методів збору матеріалів обстеження

Метод фотографії робочого дня виконавця робіт передбачає особисту участь проектувальників і застосування розрахованого для реєстрації даних спостереження спеціального листа фотографії робочого дня і розподілу його між роботами.

Метод вибіркового хронометражу окремих робіт вимагає попередньої підготовки, певних навичок і наявності спеціального секундоміра. Дані хронометражу дозволяють установити нормативи на виконання окремих операцій і зібрати докладний матеріал про техніку здійснення деяких робіт.

Метод бесід і консультацій з керівниками проводиться у формі звичайної бесіди з керівниками підприємств чи підрозділів у формі ділової консультації з фахівцями з питань, що мають глобальний характер і відносяться до визначення проблем і стратегій розвитку й управління підприємством.

Розрахунковий метод застосовується для визначення трудомісткості й вартості робіт, які підлягають виконанню за допомогою комп'ютера, а також для встановлення обсягів робіт з окремих операцій.

Метод аналогії заснований на відмові від детального обстеження якого-небудь підрозділу чи якої-небудь роботи. Використання методу вимагає наявності тотожності і не виключає загального обстеження і з'ясування таких аспектів, на які аналогія не поширюється.

У процесі вибору методу варто враховувати наступні критерії:
ступінь особистої участі проектувальника у зборі матеріалу;
часові, трудові й вартісні витрати на одержання відомостей у підрозділах.

Проектувальнику необхідно знати й у кожному конкретному випадку застосовувати найбільш економічний і такий, що забезпечує потрібну повноту відомостей, метод збору матеріалів обстеження.

10.1.5. Програма обстеження

Обстеження проводиться за заздалегідь розробленою програмою, яка складається під час виконання відповідної роботи, за формою, наведеною в табл. 7. У ній міститься перелік питань, відповіді на які дадуть повне уявлення про діяльність досліджуваного об'єкта і будуть враховані під час створення проекту ІС. Питання можна систематизувати за трьома основними напрямками дослідження об'єкта.

Фрагмент програми обстеження

№ з/п	Найменування питання	Джерело інформації	Одержувач Інформації
1	Мета функціонування об'єкта	Керівник підприємства	Керівник проекту
2	Основні параметри об'єкта	Керівник підприємства	Керівник проекту
3	Організаційна структура об'єкта	Секретар керівника	Заступник керівника проекту
.....

Перший напрямок передбачає одержання уявлення про об'єкт вивчення, тобто організаційно-економічну систему (наприклад, під-приємства) в цілому, включаючи з'ясування цілей функціонування цієї системи, виявлення значень основних параметрів діяльності підприємства і т. д.

Другий напрямок передбачає вивчення й опис організаційно-функціональної структури об'єкта (як правило, відноситься до апарату управління). При цьому вивчаються функції, виконувані в структурних підрозділах, господарські процеси і процедури, виявляються комплекси завдань, обумовлені виконуваними функціями, процесами і процедурами, визначається склад вхідної і вихідної інформації з кожного завдання.

Третій напрямок передбачає вивчення й опис структури інформаційних і (або) матеріальних потоків: складу і структури компонентів потоків, частоти їх виникнення, обсягів за встановлений період, напрямки руху потоків, процедур обробки, в яких беруть участь ці компоненти. Джерелом зведень є одержувані від фахівців предметної сфери інтерв'ю, економічна документація і результати розрахунків. Опис інформаційної структури виконується на рівні економічних документів і показників.

Для організації праці проектувальників під час виконання збору матеріалів обстеження і його наступного аналізу необхідне виконання роботи – "Розробка плану-графіка виконання", фрагмент якого наведений у табл. 8.

План-графік виконання робіт на етапі збору матеріалів обстеження

№ п/п	Найменування роботи	Код роботи	Виконавець	Дата початку	Тривалість виконання, дні	Дата закінчення
1	Визначення цілей і параметрів підприємства	001	Керівник проекту М. М. Дубов	01.10.07	2	02.10.07
2	Визначення організаційної структури підприємства	002	Заступник керівника проекту О. О. Іванов	03.10.07	1	03.10.07

"План-графік" є інструментом для планування й оперативного управління виконанням робіт на передпроектній стадії.

10.1.6. Проведення збору і формалізації матеріалів обстеження

Останньою операцією, виконуваною проектувальниками на цьому етапі, є "Проведення збору і формалізація матеріалів обстеження", у процесі яких члени бригад повинні взяти інтерв'ю у фахівців підрозділів досліджуваної предметної сфери; зібрати відомості про всі об'єкти обстеження, у тому числі про підприємство в цілому, функції управління, методи й алгоритми реалізації функцій, складанні оброблюваних показників, що розраховуються; зібрати форми документів, що відбивають господарські процеси і використовувані класифікатори, макети файлів, зведення про використовувані технічні засоби і технології обробки даних; проконтролювати разом із користувачем їхню правильність, сформулювати "Звіт про обстеження" і виконати інші роботи.

Збір матеріалів обстеження варто проводити за допомогою стандартних форм і таблиць, які зручно читати й обробляти (рис. 38). Уся одержувана документація розбивається на три групи.

До **першої групи** входять документи, що містять опис загальних

параметрів економічної системи, її організаційної структури, матричної моделі розподілу функцій, реалізованих кожним структурним підрозділом. Зокрема, загальні параметри повинні містити: найменування об'єкта і його належність (належність підприємства міністерству, об'єднанню, корпорації тощо); тип об'єкта (наприклад, тип підприємства, вид виробництва, режими роботи); види і номенклатуру продукції або послуг; види і кількість устаткування та матеріальних ресурсів; категорії і чисельність працівників і т. д.

У цю групу входить також форма опису загальних характеристик функцій управління економічною системою, господарських процесів і процедур, які реалізують ці функції. Ця форма включає відображення наступних параметрів: найменування кожної функції, процесу і процедури, опис економічної сутності завдань, розв'язуваних у процесі виконання процедури, пов'язаної з обробкою інформації; склад процедур обробки інформації, реалізованих кожним завданням; взаємозв'язок завдань, вартісні витрати, пов'язані з реалізацією кожного завдання.

Опис організаційної структури повинен включати склад і взаємозв'язок підрозділів і осіб, які реалізують функції і завдання управління. Опис виробничої структури об'єкта повинен відбивати склад і взаємозв'язок підрозділів, які реалізують виробництво товарів або послуг. Опис функціональної структури покликаний відображати розподіл функцій, господарських процесів і процедур управління між складовими організаційної структури і повинен передбачати проведення класифікації процедур, пов'язаних з обробкою даних, комунікацією між співробітниками або ухваленням управлінських рішень.

Опис матеріальних потоків передбачає відображення маршрутів руху засобів, предметів і продуктів праці, робочої сили між підрозділами виробничої структури і буде включати: опис видів продукції або послуг, ресурсів; опис технологічних операцій, їх частоту і тривалість виконання; обсяги переміщуваних ресурсів, продукції або послуг, використовувани засоби транспортування.

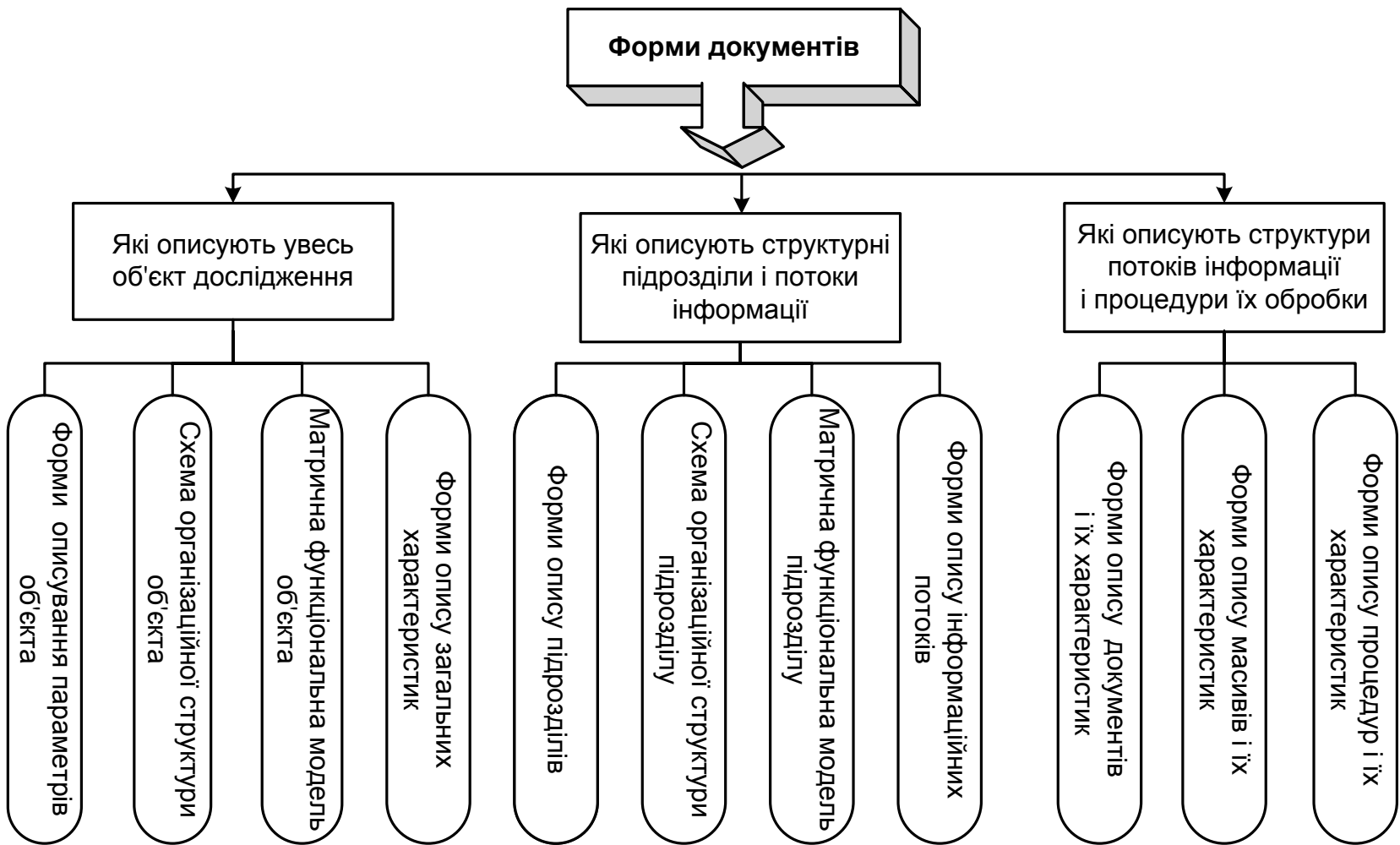


Рис. 38. **Форми документів для формалізації матеріалів обстеження**

Друга група форм формалізує матеріали обстеження для кожного структурного підрозділу, який має у своєму складі, крім форм, аналогічних тим, що входять у першу групу, форми опису інформаційних потоків за підрозділами, які здійснюють зв'язок завдань усередині кожного підрозділу між собою, а також зв'язки між підрозділами.

Форма опису документопотоків включає наступні характеристики: найменування вхідних документів, кількість їхніх екземплярів; об'ємні дані для кожного документопотоку; перелік інформаційних файлів, де використовуються ці документи; носій, на якому зберігаються дані; час створення; час використання; перелік полів файлів; вихідні документи, одержувані на основі інформації файлів.

Третя група документів містить опис компонентів кожного інформаційного потоку, включаючи документи, інформаційні файли, процедури обробки і характеристики цих компонентів.

Форми характеристик документів включають: найменування підрозділу, тип документа (первинний, проміжний чи результатний), призначення документа, найменування документа, періодичність створення або час використання. Форма опису документів містить: перелік показників; опис структури документів; перелік реквізитів; розподіл реквізитів за розділами документа; типи реквізитів.

Форма характеристик процедур обробки даних включає: найменування підрозділу, де використовується процедура, завдання, у яке входить дана процедура; вхідну інформацію, її обсяги; використовувані файли і їх обсяги; частоту звернення процедури до файла; блок-схему процедури; вихідні дані процедури.

Форма опису процедур обробки містить: найменування задачі; операції процедури; кількість операцій; використовувану техніку; вартісні та часові витрати.

Отриманий у результаті проведеної формалізації опис об'єкта містить вихідні дані для проектування ІС і визначає параметри майбутньої системи. Так, матеріальні потоки обумовлюють обсяги оброблюваної інформації, склад первинних даних, періодичність і терміни збору, їхні джерела, необхідні для розробки інформаційної бази. Функціональна структура об'єкта визначає комплекси завдань управління, що автоматизуються, для кожного з яких зазначають: склад вхідних і вихідних показників; періодичність і терміни їхнього формування; процедури використання даних показників; розподіл функцій і процедур між персоналом і технічними засобами. Організаційна структура об'єкта є підставою для виділення осіб, що визначають умову вирішення завдань обробки інформації, а також одержувачів вихідних

показників і документів.

10.1.7. Склад і зміст робіт на етапі "Аналіз матеріалів обстеження"

На основі формалізованого опису предметної сфери виконується етап "Аналіз матеріалів обстеження", метою якого є:

зіставлення всієї зібраної про об'єкт інформації з тими вимогами, що висуваються до об'єкта, визначення недоліків функціонування об'єкта обстеження;

вироблення основних напрямків удосконалювання роботи об'єкта обстеження на базі впровадження проекту ІС,

вибір напрямків проектування (вибір інструментарію) і оцінка ефективності застосування обраного інструментарію;

обґрунтування вибору рішень за основними компонентами проекту ІС і визначення загальносистемних, функціональних і локальних вимог до майбутнього проекту і його частин.

На цьому етапі виконуються наступні роботи:

аналіз і визначення складу об'єктів автоматизації;

аналіз і визначення складу завдань в кожному об'єкті, що автоматизується;

вибір комплексу технічних засобів (КТЗ);

аналіз і попередній вибір ОС;

вибір способу організації інформаційної бази (ІБ);

вибір засобу проектування ПЗ системи;

розробка ТЕО і ТЗ.

Аналіз матеріалів обстеження дозволяє проектувальникам виділити і скласти список підрозділів, які автоматизуються. На вибір об'єктів автоматизації впливає ряд факторів, наприклад, таких як:

кількість функцій, які формалізуються, у кожному конкретному підрозділі;

кількість зв'язків цього підрозділу з іншими підрозділами;

важливість цього підрозділу в процесах управління об'єктом;

ступінь підготовленості підрозділу для впровадження комп'ютерів та ін.

Згідно з цими факторами виділяють список найбільш важливих підрозділів. Наприклад, для підприємства такими підрозділами є відділи техніко-економічного планування, оперативного управління основним виробництвом, технічної підготовки виробництва, матеріально-технічного

постачання, реалізації та збуту готової продукції, бухгалтерії.

У ході виявлення списку завдань, що автоматизуються, для яких необхідно розробити проекти, проектувальники беруть до уваги наступні фактори:

- важливість вирішення завдання для виконання основних функцій управління, ділових процесів і процедур у даному підрозділі;

- трудомісткість і вартість розрахунку основних заказників даного завдання за рік;

- сильний інформаційний зв'язок розглянутого завдання з іншими завданнями;

- недостатню оперативність розрахунку показників;

- низьку вірогідність одержуваних даних;

- недостатню кількість аналітичних показників, одержуваних на базі первинних документів;

- нееквівалентний метод розрахунку показників та ін.

Крім того, на цій роботі здійснюється виявлення черг проектування розв'язуваних завдань. До завдань першої черги відносять найбільш трудомісткі завдання і завдання, що забезпечують інформацією всі інші завдання комплексів і підсистем (наприклад, завдання планування і бухгалтерського обліку).

Далі виконується робота, пов'язана з аналізом усіх отриманих раніше результатів, вхідних універсумів і попереднім вибором комплексу технічних засобів. На вибір типу комп'ютера з універсуму впливає велика кількість факторів, які прийнято поєднувати в наступні групи:

- фактори, пов'язані з параметрами вхідних інформаційних потоків, що надходять на обробку (обсяг інформації, тип носія інформації, характер подання інформації);

- фактори, що залежать від характеру завдань, які повинні вирішуватися на комп'ютері, і їхніх алгоритмів: терміновість рішення, можливість поділу завдань на підзавдання, виконувани на іншому комп'ютері, кількість файлів з умовно-постійною інформацією;

- фактори, обумовлені технічними характеристиками комп'ютера: продуктивність процесора, ємність оперативної пам'яті, підтримувана операційна система, можливість підключення різних пристроїв введення-виведення;

- фактори, що відносяться до експлуатаційних характеристик комп'ютера: необхідні умови експлуатації, необхідний штат обслуговчого персоналу і його кваліфікація;

- фактори, що враховують вартісні оцінки витрат на придбання, на

обслуговчий персонал, на проведення ремонтних робіт.

Далі виконується робота "Вибір типу операційної системи". Операційні системи здійснюють управління роботою комп'ютера, його ресурсами, запускають на виконання різні прикладні програми, виконують усілякі допоміжні дії за запитом користувача. Розрізняють однокористувальницькі, багатокористувальницькі й мережні ОС.

До факторів, що визначають вибір конкретного типу ОС і його версії, відносяться:

- необхідна кількість підтримуваних програмних продуктів;
- вимоги до апаратних засобів;
- можливість використання різних пристроїв введення-виведення;
- вимога підтримки мережної технології;
- наявність довідкової служби для користувача;
- наявність дружнього інтерфейсу і простота використання;
- можливість переконфігурації і швидкого налаштування на нові апаратні засоби;
- швидкодія;
- сумісність з іншими ОС;
- підтримка нових інформаційних технологій та ін.

Наступною роботою є "Вибір способу організації інформаційної бази". Інформаційна база має два способи організації – як сукупність локальних файлів і як інтегрована організація у вигляді баз даних. **Локальна (файлова) організація** має на увазі під собою збереження даних у вигляді сукупності локальних файлів, не залежних між собою, створюваних для документа, завдання або комплексу завдань. **Інтегрована база даних** є сукупністю взаємозалежних, що зберігаються разом даних, використовуваних для одного або декількох додатків. Дані, організовані у вигляді бази даних (БД), можуть бути організовані як централізовані бази даних, тобто розміщені на одному комп'ютері, і у вигляді розподілених БД (розміщених на декількох комп'ютерах).

Програмні засоби ведення ІБ вибираються виходячи з класу систем збереження даних: системи управління файлами або системи управління базами даних (СУБД). До основних факторів, що визначають вибір типу СУБД, відносяться наступні:

- масштаб застосування СУБД – за цією ознакою вибираються персональні – настільні СУБД (наприклад, Access) або промислові – мережні СУБД (наприклад, Oracle, Sybase, MS SQL, InterBase та ін.);
- мова спілкування: вибирають СУБД із відкритими мовами,

замкненими чи змішаними;

кількість рівнів в архітектурі: однорівневі, дворівневі, трирівневі;

виконувані СУБД функції: інформаційні – організація збереження інформації і доступу до неї та операційні функції, пов'язані з обробкою інформації;

сфера можливого застосування СУБД: універсальне використання і спеціалізоване.

У процесі виконання наступної роботи здійснюється "Вибір методів і засобів проектування програмного забезпечення системи", що прямо залежить від обраної технології проектування. В універсум методів проектування, використовуваних за канонічного підходу, входять такі, як метод структурного проектування, модульного проектування, та ін. Основними факторами, що справляють вплив на вибір методів, є їхня сумісність, скорочення часу і вартісних витрат на проектування, одержання якісного продукту, що був би зручним для наступної його експлуатації та супроводу.

Виконання всіх цих операцій завершується складанням ТЕО і формуванням ТЗ.

10.1.8. Техніко-економічне обґрунтування

Метою розробки "Техніко-економічного обґрунтування" проекту ІС є оцінка основних параметрів, що обмежують проект ІС, обґрунтування вибору й оцінка основних проектних рішень за окремими компонентами проекту. При цьому розрізняють організаційні параметри, що характеризують способи організації процесів перетворення інформації в системі, інформаційні й економічні параметри, що характеризують витрати на створення й експлуатацію системи, економію від її експлуатації. Основними об'єктами параметризації в системі є завдання, комплекси завдань, економічні показники, процеси обробки інформації.

Організаційні параметри ІС диференціюють за технологічними операціями процесу обробки інформації: збору, реєстрації, передачі, збереження, обробки й видачі інформації. Для підготовчого етапу технології обробки інформації параметрами можуть бути: вид зв'язку між джерелом інформації й комп'ютером, територіальне розміщення технічних засобів, наявність проміжного носія інформації, спосіб забезпечення вірогідності інформації тощо. Для основного етапу

технології обробки інформації в якості параметрів виступають: спосіб організації інформаційної бази, тип організації файлів, тип запам'ятовувальних пристроїв, режим обробки інформації, тип комп'ютера, тип організації використання комп'ютера і т. д. Для заключного етапу – спосіб організації зв'язку користувача з комп'ютером, наявність проміжного носія, організація розмноження результативної інформації і под.

До **інформаційних параметрів** відносяться такі, як вірогідність, періодичність збору, форма подання, періодичність обробки інформації і т. д.

До **економічних параметрів** відносяться: показники річного економічного ефекту, коефіцієнта ефективності витрат і под.

Параметризація дозволяє визначити вимоги до системи, оцінити наявну інформаційну систему, визначити придатність типових рішень у проекті ІС, вибрати проектні рішення відповідно до запропонованих вимог до ІС. До основних компонентів ТЕО належать:

- характеристика вихідних даних про предметну область;

- обґрунтування мети створення ІС;

- обґрунтування підрозділів, що автоматизуються, комплексу і, що автоматизуються, вибору комплексу технічних засобів, програмного й інформаційного забезпечення;

- розробка переліку організаційно-технічних заходів щодо проектування системи;

- розрахунок і обґрунтування ефективності обраного проекту;

- висновки про технічний рівень проекту і можливості подальших розробок.

10.1.9. Технічне завдання

На основі ТЕО розробляються основні вимоги до майбутнього проекту ІС і складається "Технічне завдання" відповідно до ГОСТ 34.602-89 "Технічне завдання на створення автоматизованої системи", до складу якого входять наступні основні розділи.

1. Загальні відомості про проект. У розділі зазначають: повне найменування системи, код системи, код договору, найменування підприємства-розробника й підприємства-замовника, перелік документів, на основі яких створюється система, планові терміни початку і закінчення

робіт зі створення системи, відомості про джерела фінансування, порядок оформлення і подання замовнику результатів робіт зі створення системи (її частин).

2. Призначення, цілі створення системи. Розділ складається з двох підрозділів:

у підрозділі "Призначення системи" зазначаються вид діяльності, що автоматизується, і перелік об'єктів автоматизації, на яких передбачається її використовувати;

у підрозділі "Цілі створення системи" вказуються найменування і необхідні значення технічних, технологічних, виробничо-економічних та інших показників об'єкта автоматизації, що будуть досягнуті в результаті впровадження ІС.

3. Характеристика об'єкта автоматизації. У розділі наводяться: короткі відомості про об'єкт автоматизації; відомості про умови експлуатації об'єкта і характеристики навколишнього середовища.

4. Вимоги до системи. Розділ складається з наступних підрозділів: вимог до системи в цілому; вимог до функцій (завдань), виконуваних системою; вимог до видів забезпечення.

У підрозділі "Вимоги до системи в цілому" вказують наступні вимоги до:

структури і функціонування системи;

чисельності та кваліфікації персоналу системи й режиму його роботи;

показників призначення;

надійності;

безпеки;

ергономіки і технічної естетики;

транспортбельності для рухомих ІС;

експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи;

захисту інформації від несанкціонованого доступу;

збереження інформації в разі аварій;

захисту від впливу зовнішніх дій;

патентної чистоти;

стандартизації та уніфікації;

та додаткові вимоги.

У підрозділі "Вимоги до функцій (завдань), виконуваних системою" для комплексів завдань і окремих завдань для кожної підсистеми наводять:

перелік функцій або завдань їхніх комплексів, що підлягають автоматизації;

розподіл їх за чергами створення;

часовий регламент реалізації кожної функції або завдання комплексу;

вимоги до якості реалізації кожної функції, завдання, комплексу, до форми подання вихідної інформації;

характеристики необхідної точності й часу виконання, вірогідності видачі результату.

У підрозділі "Вимоги до видів забезпечення" містяться вимоги до математичного, програмного, технічного, лінгвістичного, інформаційного та методичного забезпечення ІС.

5. Склад і зміст робіт зі створення системи. Розділ повинен містити: перелік стадій і етапів робіт зі створення системи відповідно до ГОСТ 34.601-90; терміни виконання; перелік організацій-виконавців; перелік документів за ГОСТ 34.201-89 "Види, комплектність і зазначення документів при створенні автоматизованих систем", запропонованих після закінчення робіт; вид і порядок проведення експертизи технічної документації та ін.

6. Порядок контролю приймання системи. У розділі вказують: види, склад, методи іспиту системи і її частин; загальні вимоги до приймання робіт за стадіями; порядок затвердження прийомних документів; статус приймальної комісії.

7. Вимоги до складу і змісту робіт з підготовки об'єкта автоматизації до введення системи в дію. У розділі необхідно навести перелік необхідних заходів і їхніх виконавців, який варто виконувати під час підготовки об'єкта до введення ІС у дію: надання інформації, що надходить у систему, вигляду, придатного для введення в комп'ютер; створення умов функціонування об'єкта, за яких гарантується відповідність створюваної системи вимогам, що містяться в ТЗ; створення необхідних для функціонування системи підрозділів і служб; встановлення термінів і порядку комплектування штатів і навчання персоналу.

8. Вимоги до документування. У розділі наводять: перелік комплектів і видів документів, які підлягають розробці й відповідають вимогам ДСТ 34.201-89.

9. Джерела розробки. У розділі повинні бути перераховані документи й інформаційні матеріали (ТЕО, звіти про закінчені науково-дослідні

розробки, інформаційні матеріали на вітчизняні, закордонні системи-аналоги та ін.).

До складу ТЗ за наявності затверджених методик включають додатки, що містять розрахунки економічної ефективності системи; оцінку науково-технічного рівня системи.

10.2. Технологія техноробочого проектування

10.2.1. Склад і зміст робіт на етапі технічного проектування

Роботи на стадії "Техноробочого проектування" виконуються на основі затвердженого "Технічного завдання". Розробляються основні положення проектованої системи, принципи її функціонування і взаємодії з іншими системами; визначається структура системи; розробляються проектні рішення щодо забезпечувальних частин системи.

На етапі "Технічне проектування" здійснюється логічна проробка функціональної та системної архітектури ІС, у процесі якої будуються кілька варіантів усіх компонентів системи; проводиться оцінка варіантів за показниками: вартості, трудомісткості, вірогідності одержуваних результатів, і складається "Технічний проект" (ТП) системи. Усі роботи першого етапу можна розбити на дві групи.

До першої групи відноситься **розробка загальносистемних проектних рішень**, у тому числі:

- розробка загальносистемних положень щодо ІС;
- зміна організаційної структури;
- визначення функціональної структури;
- розробка проектно-кошторисної документації і розрахунків економічної ефективності системи;
- розробка плану заходів щодо впровадження ІС.

У процесі розробки основних положень щодо системи уточнюється мета створення ІС і виконувани нею функції; встановлюється її взаємозв'язок з іншими системами і формується документ "Основні положення". Далі уточнюється і змінюється організаційна структура і формується опис організаційної структури.

Найбільш принциповою в даному комплексі робіт є розробка функціональної архітектури ІС на базі універсуму принципів виділення функціональних підсистем (модулів, контурів): предметного, функціонального, змішаного (предметно-функціонального) і проблемного.

До другої групи робіт, виконуваних на етапі технічного проектування, відносяться **розробки локальних проектних рішень**, до числа яких належать наступні роботи:

розробка "Постановки завдання" для завдань, що входять до складу кожної функціональної підсистеми, яка включає основні компоненти опису завдання, і розробки, що є підставою для проектних рішень щодо завдання;

розгляд матеріалів про прив'язку ТПР у розрізі кожної функціональної підсистеми;

розробка положення про забезпечувальні підсистеми (інформаційне, технічне, математичне, програмне, лінгвістичне й організаційно-правове забезпечення);

розрахунок економічної ефективності від упровадження системи (обґрунтування);

складання плану заходів щодо підготовки об'єкта до впровадження проекту ІС.

Результатом робіт на даній стадії є затверджений "Технічний проект", склад і зміст якого регламентуються стандартом (ГОСТ 34.201-89). Документація ТП передається на експертизу, узгоджується із замовником, а потім затверджується.

10.2.2. Постановка комплексу завдань

"Опис постановки завдання (комплексу завдань)" – основний документ з локальними проектними рішеннями, що є базою для розробки інформаційного, програмного і технологічного забезпечення для кожного завдання. Відповідно до РД 50-34.698-90 [47], документ "Опис постановки завдання (комплексу завдань)" містить розділи:

характеристика комплексу завдань;

вихідна інформація;

вхідна інформація.

У розділ "**Характеристика комплексу завдань**" включають сім пунктів:

1) призначення комплексу завдань;

2) перелік об'єктів (технологічних об'єктів управління, підрозділів підприємства і т. д.), у процесі управління якими вирішують комплекс завдань;

3) періодичність і тривалість вирішення;

4) умови, за яких припиняється вирішення комплексу завдань автоматизованим способом (за необхідності);

5) зв'язки даного комплексу завдань з іншими комплексами (завданнями) ІС;

6) посади осіб і (або) найменування підрозділів, що визначають умови і часові характеристики конкретного вирішення завдання (якщо вони не визначені загальним алгоритмом функціонування системи);

7) розподіл дій між персоналом і технічними засобами в різних ситуаціях вирішення комплексу завдань.

Розділ "Вихідна інформація" містить:

1) перелік і опис вихідних повідомлень;

2) перелік і опис структурних одиниць інформації вихідних повідомлень (показників, реквізитів і їх сукупностей, сигналів управління), що мають самостійне смислове значення або посилання на документи, що містять ці дані.

Розділ "**Вихідна інформація**" включає два пункти: "Перелік і опис вихідних повідомлень" і "Перелік і опис структурних одиниць інформації вихідних повідомлень".

У пункті "**Перелік і опис вихідних повідомлень**" необхідно навести їхній опис за формою табл. 9.

Код повідомлення для машинограми (МГ) або відеокадру (ВК) повинен містити 7 знаків відповідно до наступної структури (рис. 39).

Таблиця 9

Перелік і опис вихідних повідомлень

Ідентифікатор (код повідомлення)	Найменування вихідного повідомлення	Форма подання (МГ, ВК, масив)	Періодичність видачі	Термін видачі і припустимий час затримки вирішення	Одержувачі	Призначення вихідної інформації

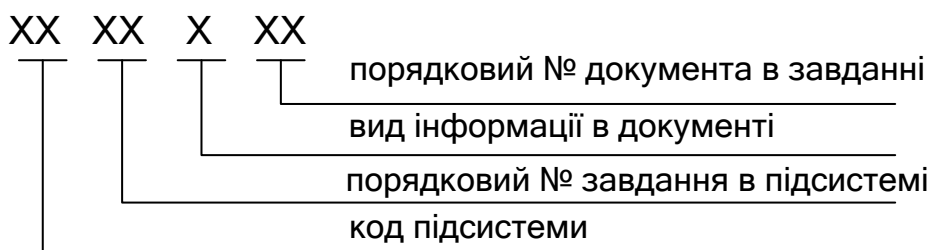


Рис. 39. Структура коду повідомлення

Коди для видів інформації подані в табл. 10.

Таблиця 10

Коди видів інформації

Вид інформації	Вхідна	Збережена	Вихідна
Нормативна	1	2	3
Планова	4	5	6
Облікова, контролю, аналізу	7	8	9

Для вихідних масивів, сформованих у завдання, наводиться літерний ідентифікатор, а не цифровий код.

У пункті "Перелік і опис структурних одиниць інформації вихідних повідомлень" необхідно описати показники, реквізити, сигнали управління, що мають самостійне суттєве значення, за формою табл. 11.

Таблиця 11

Перелік і опис структурних одиниць вихідних повідомлень

Найменування структурної одиниці інформації	Ідентифікатор	Вимоги до точності

Опис структурних одиниць у табл. 11 необхідно навести для кожного вихідного повідомлення, в тому числі описати реквізити, які розміщуються в заголовку МГ або ВК.

Розділ "**Вхідна інформація**" повинен містити:

- 1) перелік і опис вхідних повідомлень;
- 2) перелік і опис структурних одиниць інформації вхідних повідомлень або посилання на документи, що містять ці дані.

У пункті "**Перелік і опис вхідних повідомлень**" необхідно навести перелік і опис вхідних документів (за формою табл. 12) і вхідних масивів для вирішення завдання (за формою табл. 13).

Таблиця 12

Перелік і опис вхідних документів

Ідентифікатор документа	Найменування документа	Термін надходження	Частота надходження	Постачальник документа

Характеристика масивів вхідної інформації

Найменування масиву	Ідентифікатор (ім'я) масиву	Тип масиву	Технологія формування

У пункті "Перелік і опис структурних одиниць інформації вхідних повідомлень" необхідно навести опис для кожної структурної одиниці інформації вхідних повідомлень за формою табл. 14.

Таблиця 14

Перелік і опис структурних одиниць вхідних документів

Найменування структурної одиниці	Точність числового значення	Джерела інформації (документ)	Ідентифікатор джерела інформації

10.2.3. Склад і зміст робіт на етапі робочого проектування

На другому етапі – "Робоче проектування" – здійснюється технічна реалізація обраних найкращих варіантів і розробляється документація "Робочий проект" (РП). Розробка РП – завершальний етап на стадії проектування ІС. Цей проект розробляється на основі затверджених ТЗ та ТП і затвердженню не підлягає.

Мета робочого проектування – складання технічної документації для відлагодження та впровадження ІС, проведення Приймально-здавальних досліджень, а також для забезпечення нормального функціонування ІС.

На цьому етапі виконуються наступні роботи:

розробка робочої документації на систему або її частини;

розробка або адаптація програм.

Під час **розробки робочої документації на систему або її частини** здійснюють розробку робочої документації, що містить всі необхідні і достатні відомості для забезпечення виконання робіт щодо введення ІС в дію і її експлуатації, а також для підтримки рівня експлуатаційних характеристик (якості) системи відповідно до ухвалених проектних рішень, її оформлення, узгодження і затвердження. До переліку видів документів, що розробляються, входять документи відповідно до ГОСТ

34.201-89. Це технологічні інструкції з обробки даних, інструкції для роботи в умовах функціонування розробленої системи, довідник користувача.

У процесі розробки робочої документації вирішуються наступні завдання:

- ухвалення рішень щодо організації розробки робочої документації;
- складання загальносистемних проектних рішень;

розробка проектної документації щодо забезпечувальних підсистем. Цей процес розпочинається на етапі технічного проектування ІС, а завершується для деяких підсистем на етапі її робочого проектування. Під час розробки положень про забезпечувальні підсистеми враховуються дані, зібрані в процесі обстеження об'єкта та його ІС;

оформлення, погодження, затвердження робочої документації у встановленому порядку.

Під час **розробки або адаптації програм** проводять розробку програм і програмних засобів системи, вибір, адаптацію і (або) прив'язку програмних засобів, розробку програмної документації відповідно до ГОСТ 19.101.

У ході розробки робочої документації на її замовника покладаються такі завдання:

- забезпечення приймання системи в дослідну експлуатацію;

- завершення формування інформаційної бази та організація її експлуатації;

- передача на вимогу розробника даних, необхідних для тестування програмного забезпечення на контрольних прикладах;

- організація приймання програм;

- виконання основних організаційно-технічних заходів щодо підготовки підприємства до введення ІС в дію.

Розробник розробляє та оформляє технічну документацію з усіх видів забезпечень, тестує програмне забезпечення.

За результатами виконання перерахованих робіт складається документація РП згідно з ГОСТ.

10.2.4. Зміст проектної документації на інформаційні системи

Проектна документація ТП системи складається із семи частин.

1. Загальносистемна документація:

пояснювальна записка – короткий зміст проекту із зазначенням даних про проведені узгодження відповідно до чинних норм і правил; план заходів щодо підготовки об'єкта до введення ІС в експлуатацію; розрахунок економічної ефективності.

2. Документація функціональної частини:

схема функціональної структури ІС;
опис автоматизованих функцій;
опис постановки завдання (комплексу завдань).

3. Документація організаційного забезпечення:

схема організаційної структури;
опис організаційної структури.

4. Документація інформаційного забезпечення:

опис інформаційного забезпечення ІС;
опис систем класифікації та кодування економічної інформації;
опис організації інформаційної бази;
опис масивів інформації;
креслення форм документів (відеокадрів);
перелік вхідних сигналів і даних;
перелік вихідних сигналів.

5. Документація технічного забезпечення:

проектне оцінювання (обґрунтування) комплексу технічних засобів (КТЗ);

технічні вимоги (завдання) до технологічного об'єкта управління (ОУ);
відомість обладнання та матеріалів (перелік);
опис КТЗ (схема);
план розміщення КТЗ (креслення).

6. Документація математичного забезпечення (опис алгоритмів).

7. Документація ПЗ (його опис).

Проектна документація РП системи складається із п'яти частин.

1. Загальносистемна документація:

загальний опис;
відомість документів робочого (техноробочого) проекту.

2. Документація інформаційного забезпечення (опис технологічного процесу обробки даних).

3. Документація організаційного забезпечення (посадові та технологічні інструкції, а також інструкції користувача).

4. Документація технічного забезпечення:
креслення загального вигляду;
креслення встановлення технічних засобів;
замовна специфікація (перелік);
таблиця з'єднань, підключень.

5. Документація ПЗ:
детальний опис програми;
текст програми з коментарями;
контрольний приклад із даними.

10.3. Технологія впровадження, експлуатації і супроводу проекту

10.3.1. Склад і зміст робіт на стадії впровадження проекту

На стадії "Впровадження проекту" проводиться підготовка і поступове освоєння розробленої проектної документації ІС замовниками системи. У процесі виконання робіт на цій стадії здійснюється виявлення часткових і принципових системних недоробок у запропонованому для впровадження проектному рішенні.

Впровадження може здійснюватися з використанням наступних методів:

послідовний метод, коли послідовно впроваджується одна підсистема за іншою і одне завдання йде за іншим завданням;

паралельний метод, за якого всі завдання впроваджуються у всіх підсистемах одночасно;

змішаний підхід, відповідно до якого проектувальники, впровадивши кілька підсистем першим методом і накопичивши досвід, розпочинають паралельне впровадження інших.

Недоліком першого підходу є збільшення тривалості впровадження, що веде за собою зростання вартості проекту. У разі використання другого підходу скорочується час упровадження, але виникає можливість пропуску помилок у проектній документації, тому найчастіше використовують змішаний метод впровадження проекту ІС.

Впровадження проекту здійснюється протягом трьох етапів:

підготовки об'єкта до впровадження;
дослідного впровадження;
здачі проекту в промислову експлуатацію.

На етапі "**Підготовка об'єкта до впровадження**" виконуються

наступні роботи:

- змінюється організаційна структура об'єкта (підприємства);
- набираються кадри відповідної кваліфікації в галузі обробки інформації та експлуатації системи і супроводу проектної документації;
- обладнується будівля під встановлення обчислювальної техніки;
- виконуються закупівля і встановлення обчислювальної техніки з периферією;

- у цехах, відділах установлюються засоби збору, реєстрації первинної інформації і передачі каналами зв'язку;

- здійснюється встановлення каналів зв'язку;

- проводиться розробка нових документів і класифікаторів;

- здійснюється створення файлів інформаційної бази з нормативно-довідковою інформацією.

На вхід цього етапу надходять компоненти "Технічного проекту" у частині "Плану заходів щодо впровадження", рішення з технічного й інформаційного забезпечення, технологічні й інструкційні матеріали "Робочого проекту". У результаті виконання етапу складається "Акт готовності об'єкта до впровадження" проекту ІС. Потім формується склад приймальної комісії, розробляється "Програма проведення дослідного впровадження" і видається "Наказ про початок дослідного впровадження".

На етапі **"Дослідне впровадження"** впроваджуються проекти декількох завдань у декількох підсистемах. У процесі дослідного впровадження виконуються наступні роботи:

- підготовка вихідних оперативних даних для завдань, що проходять дослідну експлуатацію;

- уведення вхідних даних в комп'ютер і виконання запланованого числа реалізації вирішення завдань;

- аналіз результатних даних на предмет наявності помилок.

У випадку виявлення помилок здійснюються пошук причин і джерел помилок, внесення корективів у програми, в технологію обробки інформації, у роботу технічних засобів, у вихідні оперативні дані й у файли з умовно-постійною інформацією. Крім того, виявляється некваліфікована робота операторів, що є основою для проведення комплексу заходів для поліпшення підготовки кадрів.

Після усунення помилок одержують "Акт про проведення дослідного впровадження", що слугує сигналом для початку виконання

наступного етапу.

На третьому етапі **"Здача проекту в промислову експлуатацію"** використовують наступну сукупність документів:

договірну документацію;

"Наказ про розроблення ІС";

ТЕО і ТЗ;

виправлений "Техноробочий проект";

"Наказ про початок промислового впровадження";

"Програма проведення іспитів";

"Вимоги до науково-технічного рівня проекту системи".

У процесі здачі проекту в промислову експлуатацію здійснюються наступні роботи:

перевірка відповідності виконаної роботи договірній документації за часом виконання, обсягом проробленої роботи і витратами коштів;

перевірка відповідності проектних рішень щодо ІС вимогам ТЗ;

перевірка відповідності проектної документації щодо ГОСТ;

перевірка технологічних процесів обробки даних щодо всіх завдань і підсистем;

перевірка якості функціонування інформаційної бази, оперативності і повноти відповідей на запити;

виявлення локальних і системних помилок і їхнє виправлення.

Крім того, приймальна комісія визначає науково-технічний рівень проекту і можливості розширення проектних рішень за рахунок включення нових компонентів. У результаті виконання робіт на даному етапі здійснюється доопрацювання "Техноробочого проекту" за рахунок виявлення системних і локальних помилок і складається "Акт здачі проекту в промислову експлуатацію".

10.3.2. Склад і зміст робіт на стадії "Експлуатація і супровід проекту"

На четвертій стадії – "Експлуатація і супровід проекту" – виконуються наступні етапи:

експлуатація проекту;

супровід і модернізація проекту.

На цій стадії вирішується питання про те, чиїми силами (персоналом замовника чи організації-розробника) будуть здійснюватися експлуатація і супровід проекту, і у випадку вибору другого варіанта складається "Договір про супровід проекту".

У процесі виконання етапу **"Експлуатація проекту"** здійснюються виправлення в роботі всіх частин системи в разі виникнення збоїв, реє-

страція цих випадків у журналах, відстеження техніко-економічних характеристик роботи системи і накопичення статистики про якість роботи всіх компонентів системи.

На етапі "**Супровід і модернізація проекту**" виконується аналіз зібраного статистичного матеріалу, а також аналіз відповідності параметрів роботи системи вимогам навколишнього середовища. Аналіз здійснює створювана для цих цілей комісія. Результати аналізу дозволяють:

зробити висновок про необхідність модернізації всього проекту чи його частин;

визначити обсяги доробок, терміни і вартість виконання цих робіт з метою одержання "Техноробочого проекту", що пройшов модернізацію. У випадку виявлення факту морального старіння проекту комісією ухвалюється рішення про доцільність проведення його утилізації чи розробки нового проекту для даного об'єкта.

З метою організації робіт щодо супроводу, модернізації та розвитку ІС необхідно під час її створення керуватися принципами системності, розвитку, сумісності, стандартизації й уніфікації, ефективності, що дають змогу поліпшити функціонування ІС.

Принцип **системності** полягає в тому, що під час створення, функціонування і розвитку системи мають бути встановлені та збережені зв'язки між її структурними елементами, які забезпечують цілісність ІС.

Принцип **розвитку** означає, що ІС має створюватися з урахуванням її можливого заповнення й оновлення функцій, видів її забезпечень доопрацюванням програмних технічних засобів.

Принцип **сумісності** полягає в забезпеченні взаємодії ІС різних рівнів і видів у процесі їх спільного функціонування.

Принцип **стандартизації та уніфікації** – це раціональне застосування типових уніфікованих і стандартизованих елементів під час створення та розвитку ІС.

Принцип **ефективності** полягає в досягненні раціонального співвідношення між витратами на створення ІС і цільовими ефектами, одержуваними під час її функціонування.

Процеси функціонування, супроводу та модернізації проекту ІС відбуваються, як правило, паралельно.

Під час впровадження і супроводу системи робота зводиться до усунення можливих помилок в інформаційному та програмному забезпеченні, у технології обробки даних або до зміни функціональної частини ІС. По ефективніших сучасних засобів обчислювальної техніки і

новітніх інформаційних технологій (ІТ) веде до модернізації ІС.

Роботи із супроводу проекту ІС або з його модернізації спричиняють трудові й вартісні витрати. Зменшення їх залежить від методології виконання проектних робіт, гнучкості створеної системи.

Правильне використання методології проектування значно зменшує витрати на супровід розробленої ІС. Якість методології проектування визначається правильністю виконання:

- декомпозиції системи;
- декомпозиції процесу її проектування;
- опису процесу проектування ІС мовними засобами.

Розвиток ІС – це процес розширення складу її функцій, що ґрунтується на результатах аналізу функціонування ІС й об'єкта, спрямованих на підвищення ефективності діяльності останнього. Розвиток проекту ІС за завданням замовника здійснюється доопрацюванням програмного, інформаційного та технічного забезпечення. Допустима межа модернізації, а також розвитку системи має бути встановлена в ТЗ під час розробки розділу "Вимоги до системи".

У процесі створення, функціонування і розвитку ІС потрібно оцінювати її науково-технічний рівень із метою перевірки відповідності систем останнім досягненням науки й техніки.

Використовуючи єдиний підхід до проектування та супроводу ІС, можна значно послабити процес її руйнації під час функціонування, тобто ЖЦ системи подовжується приблизно на 50%.

10.3.3. Приймально-здавальна документація на ІС

До приймально-здавальної документації на ІС належать:

1) акт завершення робіт – фіксує факт завершення окремої роботи та її результати під час створення ІС. Цей документ не поширюється на будівельно-монтажні, а також пусканалагоджувальні роботи;

2) план-графік робіт – установлює перелік робіт, пов'язаних зі створенням ІС, їх терміни, виконавців робіт;

3) наказ про виконання робіт – залежно від етапу робіт документ має такі різновиди:

- наказ про готовність ОУ до проведення будівельно-монтажних робіт;
- наказ про готовність ОУ до проведення пусканалагоджувальних

робіт;

наказ про початок випробної експлуатації ІС;

наказ про введення ІС у промислову експлуатацію;

4) наказ про склад приймальної комісії – висвітлює склад і підставу для організації комісії, найменування організацій (розробника, співвиконавців, замовника), призначення та цілі комісії, терміни початку і закінчення роботи, форму її завершення;

5) програма робіт та її різновиди:

програма випробування;

програма випробної експлуатації;

програма роботи приймальної комісії;

6) протокол випробувань – фіксує результати попередніх випробувань під час передачі ІС загалом або її частин у дослідну чи промислову експлуатацію;

7) акт приймання у випробну експлуатацію – фіксує факт завершення введення ІС загалом або її частин у випробну експлуатацію. У документі відображаються основні результати приймання і рішення комісії про прийняття ІС у випробну експлуатацію;

8) акт приймання у промислову експлуатацію – фіксує факт введення ІС загалом або її частин у промислову експлуатацію. До цього документа додаються:

програма і протокол випробувань;

протокол засідань комісії;

акти приймання у промислову експлуатацію прийнятих раніше частин ІС;

перелік технічних засобів, які використовувала комісія під час приймання ІС;

9) протокол узгоджень – призначений для узгодження відхилень від раніше ухвалених та затверджених проектних рішень (у ТЗ, ТП), коли в процесі випробної експлуатації ІС загалом або її частин виникла необхідність їх коригування.

Під час упровадження ІС оцінюють її споживчі властивості, а також ефективність (за відомими методиками).

Контрольні запитання

1. Що таке канонічне проектування? Назвіть його стадії.
2. Назвіть найважливіші об'єкти обстеження в процесі створення ІС.
3. Які етапи виділяють на передпроектній стадії?
4. Які роботи виконуються під час збору матеріалів обстеження? Розкрийте їх зміст.
5. Наведіть і охарактеризуйте методи обстеження на передпроектній стадії.
6. Наведіть і охарактеризуйте методи збору матеріалів обстеження.
7. Який зміст програми обстеження?
8. Які документи використовуються для формалізації матеріалів обстеження?
9. Який склад і зміст робіт виконуваних у процесі аналізу матеріалів обстеження?
10. Яка технічна документація складається після виконання передпроектного обстеження? Який її зміст?
11. Яке призначення стадії "Техноробоче проектування"?
12. Який склад і зміст робіт на етапі технічного проектування ІС?
13. Яка структура і зміст документа "Постановка завдання"?
14. Який склад та зміст робіт на етапі робочого проектування ІС?
15. Яка документація розробляється на стадії "Техноробоче проектування"?
16. Який склад, послідовність і зміст робіт на стадії "Впровадження проекту"?
17. Яка документація складається на стадії "Впровадження проекту"?
18. Які роботи виконуються під час експлуатації, супроводу та модернізації проекту?

Модуль 4. Індустріальні методи проектування ІС

11. Підходи до проектування ІС

11.1. Методології моделювання предметної області

11.1.1. Система моделей предметної області

Головна особливість проектування ІС обумовлена складністю робіт, що виконуються на фазах аналізу і проектування за відносно невисокої складності і трудомісткості наступних фаз життєвого циклу. Крім того, невирішені питання і помилки, що мали місце під час аналізу і проектування, зароджують на подальших фазах важкі, часто нерозв'язні проблеми і, нарешті, можуть позбавити успіху.

Тому для успішної реалізації проекту ІС об'єкт проектування повинен бути передусім адекватно описаний, повинні бути побудовані повні й несуперечливі моделі предметної сфери з використанням сучасних інструментальних засобів.

Сучасні підходи до проектування є сукупністю методів моделювання складних систем з великою розмірністю вирішуваних завдань. Внаслідок цього вони повинні спиратися на засоби комп'ютерної підтримки, що забезпечують автоматизацію праці системних аналітиків. Такими засобами є CASE-системи.

Архітектура більшості CASE-систем включає методологію, моделі, нотації і засоби, які можна визначити таким чином (рис. 40):

методологія представляє методи і засоби для дослідження структури і діяльності організації. Вона визначає основні принципи і прийоми використання моделей;

модель становить сукупність символів (графічних, математичних і т. д.), яка адекватно описує деякі властивості модельованого об'єкта і відношення між ними;

нотація – це система умовних позначень, прийнята в конкретній моделі;

засоби – апаратне і програмне забезпечення, що реалізує вибрану методологію, зокрема побудову відповідних моделей з прийнятою для них нотацією.

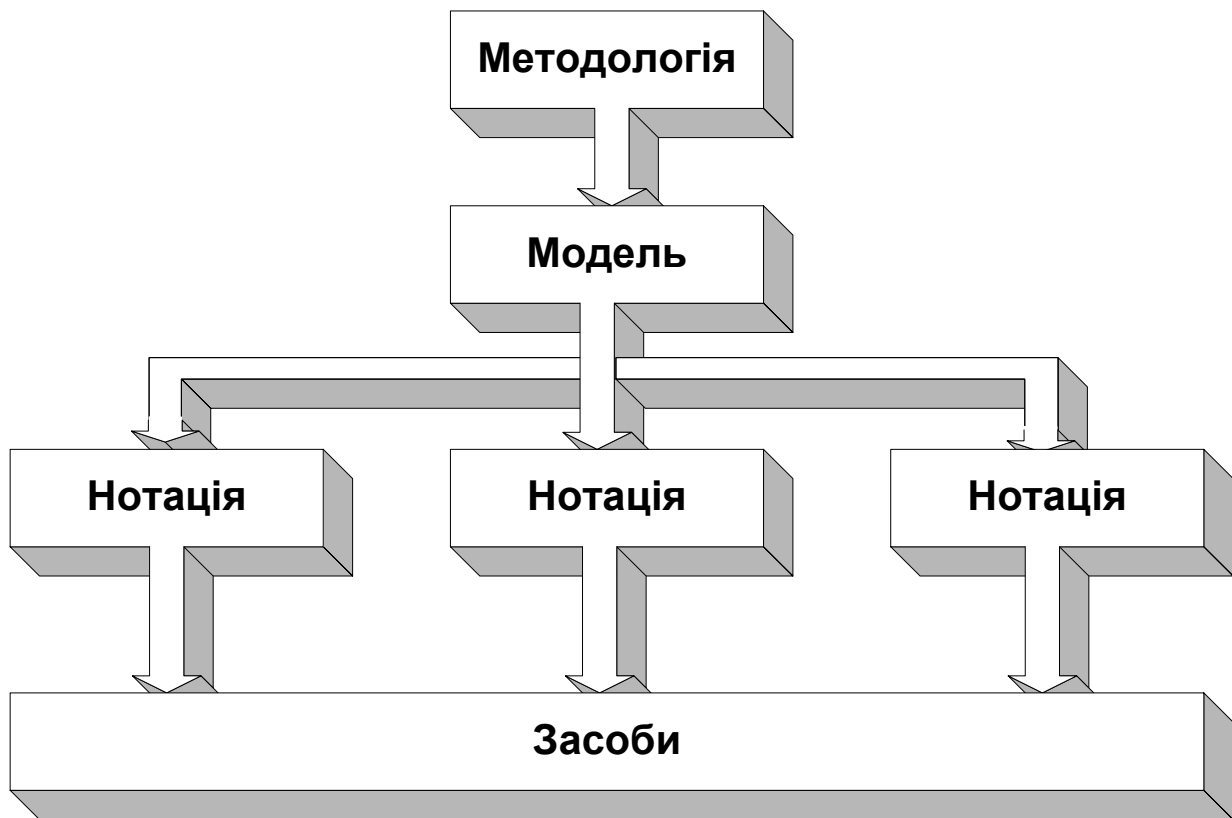


Рис. 40. Архітектура CASE-систем

У сучасних підходах до проектування ІС в поняття предметної сфери вкладається більш широкий смисл, ніж просто сукупність керованих об'єктів підприємства. **Предметну область** розглядають як взаємозалежну сукупність:

- керованих об'єктів;
- суб'єктів управління;
- функцій управління, що автоматизуються;
- програмно-технічних засобів їх реалізації.

Необхідність моделювання предметної області багато в чому обумовлена складністю як інформаційної системи, так і самої організації з функціональної і системної точок зору.

Для того щоб одержати адекватний предметній області проект ІС у вигляді системи програм, які правильно працюють, необхідно мати цілісне, системне подання моделі, що відбиває всі аспекти функціонування майбутньої інформаційної системи. При цьому під **моделлю** мається на увазі певна система, що імітує структуру або функціонування досліджуваної предметної області, яка відповідає основній вимозі – адекватності цієї сфери.

Проведення попереднього моделювання предметної області

дозволяє скоротити час і терміни проведення проектувальних робіт і одержати більш ефективний і якісний проект. Без проведення моделювання предметної області велика ймовірність одержання неякісної ІС, у якій може бути допущена велика кількість помилок у вирішенні стратегічних питань, що приводять до економічних втрат і високих витрат на наступне перепроектування системи. Унаслідок цього всі сучасні технології проектування ІС ґрунтуються на використанні методології моделювання предметної області. Моделі дають можливість оцінити переваги і недоліки наявної інформаційної системи підприємства і побудувати ефективну архітектуру нової інформаційної системи.

До моделей предметних галузей висуваються наступні **вимоги**:

формалізованість, що забезпечує однозначний опис структури предметної області. Для подання моделей використовуються нотації різних формальних мов моделювання;

зрозумілість для замовників і розробників, що досягається на основі застосування графічних засобів відображення моделі;

можливість реалізувати, що передбачає наявність засобів фізичної реалізації моделі предметної області в ІС;

оцінка ефективності реалізації моделі предметної області на основі визначених методів і показників, що обчислюються.

Для реалізації перерахованих вимог, як правило, будується система моделей, що відбиває структурний і оцінний аспекти функціонування предметної області.

Структурний аспект функціонування ІС передбачає побудову:

об'єктної структури, що відбиває склад матеріальних і інформаційних об'єктів предметної області, що взаємодіють у процесах;

функціональної структури, що відбиває взаємозв'язок функцій (дій) щодо перетворення об'єктів у процесах;

структури управління, що відбиває події і бізнес-правила, які впливають на виконання процесів;

організаційної структури, що відбиває взаємодію організаційних одиниць підприємства і персоналу в процесах;

технічної структури, що описує топологію розташування і способи комунікації комплексу технічних засобів.

Для подання структурного аспекту моделей предметних областях в основному використовуються графічні методи, що повинні гарантувати подання інформації про компоненти системи. Головна вимога до графічних методів документування – простота. Графічні методи повинні

забезпечувати можливість структурної декомпозиції специфікацій системи з максимальним ступенем деталізації й узгодженням описів на суміжних рівнях декомпозиції.

Безпосередньо з моделюванням пов'язана проблема вибору мови подання проектних рішень (нотації), що дозволяє якнайбільше залучати майбутніх користувачів системи до її розробки. Ця мова, з одного боку, повинна робити рішення проектувальників зрозумілими користувачу, з іншого боку, надавати проектувальникам засоби досить формалізованого й однозначного визначення проектних рішень, які підлягають реалізації у вигляді програмних комплексів, що утворюють цілісну систему програмного забезпечення.

Графічне зображення нерідко виявляється найбільш ємною формою подання інформації. При цьому проектувальники повинні враховувати, що графічні методи документування не можуть повністю забезпечити декомпозицію проектних рішень від постановки завдання проектування до реалізації програм комп'ютерів. Труднощі виникають при переході від етапу аналізу системи до етапу проектування й особливо до програмування. Окрема програма може не бути результатом прямої декомпозиції деякої функції системи: вона може виконувати певну обробку інформації для декількох функцій у системі.

Головний критерій адекватності структурної моделі предметній області полягає у функціональній повноті розроблюваної ІС.

Оцінні аспекти моделювання предметної області пов'язані з розроблюваними показниками ефективності процесів, що автоматизуються, до яких відносяться:

час вирішення завдань;

вартісні витрати на обробку даних;

надійність процесів;

непрямі показники ефективності, такі як обсяги виробництва, продуктивність праці, оборотність капіталу, рентабельність і т. д.

Для розрахунку показників ефективності ІС, що реалізує модель предметної області, як правило, використовуються статистичні методи функціонально-вартісного аналізу (АВС-аналізу) і динамічні методи імітаційного моделювання.

11.1.2. Рівні деталізації проблемної області

В основі різних методологій моделювання предметних областей ІС лежать принципи послідовної деталізації абстрактних категорій. Звичайно моделі будуються на трьох рівнях:

зовнішньому рівні (визначення вимог);
концептуальному рівні (специфікації вимог);
внутрішньому рівні (реалізації вимог).

На **зовнішньому** рівні модель відповідає на запитання, **що повинна робити система**, тобто визначається склад основних компонентів системи: об'єктів, функцій, подій, організаційних одиниць, технічних засобів. На **концептуальному** рівні модель відповідає на запитання, **як повинна функціонувати система**, тобто визначається характер взаємодії компонентів системи одного і різних типів. На **внутрішньому** рівні модель відповідає на запитання, **за допомогою яких програмно-технічних засобів реалізуються вимоги до системи?** З позиції життєвого циклу ІС описані рівні моделей відповідно будуються на етапах аналізу вимог, логічного (технічного) і фізичного (робочого) проектування.

Слід розглянути більш детально особливості побудови моделей предметної області на трьох рівнях деталізації.

1. Об'єктна структура.

Об'єкт – це сутність, що використовується під час виконання певної функції або операції (перетворення, обробки, формування і т. д.). Об'єкти можуть мати динамічну або статичну природу: динамічні об'єкти використовуються в одному циклі відтворення, наприклад замовлення на продукцію, рахунки на оплату, платежі; статичні об'єкти використовуються в багатьох циклах відтворення, наприклад устаткування, персонал, запаси матеріалів.

На зовнішньому рівні деталізації моделі виділяються основні види матеріальних об'єктів (наприклад, сировина і матеріали, напівфабрикати, готові вироби, послуги) і основні види інформаційних об'єктів чи документи (наприклад, замовлення, накладні, рахунки і т. д.).

На концептуальному рівні побудови моделі предметної області уточнюється склад класів об'єктів, визначаються їх атрибутивний склад і взаємозв'язки між собою. Таким чином, будується узагальнене подання структури предметної області.

Далі концептуальна модель на внутрішньому рівні відображається у вигляді файлів бази даних, вхідних і вихідних документів ІС. Причому динамічні об'єкти подаються одиницями змінної інформації чи документами, а статичні об'єкти – одиницями умовно-постійної інформації у вигляді списків, номенклатури, цінників, довідників, класифікаторів. Модель бази даних як постійно підтримуваного

інформаційного ресурсу відображає збереження умовно-постійної і змінної інформації, що накопичується, використовуваної в повторюваних інформаційних процесах.

2. Функціональна структура.

Функція (операція) становить певний перетворювач вхідних об'єктів у вихідні. Послідовність взаємозалежних за входами і виходами функцій становить бізнес-процес. Функція бізнес-процесу може породжувати об'єкти будь-якої природи (матеріальні, грошові, інформаційні). Причому бізнес-процеси та інформаційні процеси, як правило, нерозривні, тобто функції матеріального процесу не можуть здійснюватися без інформаційної підтримки. Наприклад, функція відвантаження готової продукції здійснюється на основі документа "Замовлення", що, у свою чергу, породжує документ "Накладна", який супроводжує партію відвантаженого товару.

Функція може бути представлена однією дією чи певною сукупністю дій. В останньому випадку кожній функції може відповідати певний процес, у якому підфункціям можуть відповідати свої підпроцеси і т. д., доки кожна з підфункцій не буде представляти деяку недекомповану послідовність дій.

На зовнішньому рівні моделювання визначається список основних функцій чи видів бізнес-процесів. Звичайно таких функцій нараховується 15 – 20.

На концептуальному рівні виділені функції декомпонуються і будуються ієрархії взаємозалежних функцій.

На внутрішньому рівні відображається структура інформаційного процесу в комп'ютері: визначаються ієрархічні структури програмних модулів, які реалізують функції, що автоматизуються.

3. Структура управління.

У сукупності функцій бізнес-процесу можливі альтернативні чи циклічні послідовності залежно від різних умов перебігу процесу. Ці умови пов'язані з діями, що відбуваються, у зовнішньому середовищі чи в самих процесах і утворенням визначених станів об'єктів (наприклад, замовлення прийняте, відкинута, відправлене на коректування). **Події** викликають виконання функцій, які, у свою чергу, змінюють стани об'єктів і формують нові події і т. д., поки не буде завершений певний бізнес-процес. Тоді послідовність подій складає конкретну реалізацію бізнес-процесу.

Кожна дія описується з двох точок зору: інформаційної і

процедурної. Інформаційно дія відбивається у вигляді певного повідомлення, що фіксує факт виконання певної функції зміни стану чи заяви нового об'єкта. Процедурно дія викликає виконання нової функції, і тому для кожного стану об'єкта повинні бути задані описи цих викликів. Таким чином, події виступають у сполучній ролі для виконання функцій бізнес-процесів.

На зовнішньому рівні визначається список зовнішніх подій, які викликаються взаємодією підприємства із зовнішнім середовищем (платежі податків, відсотків за кредитами, постачання за контрактами і т. д.), і список цільових настанов, яким повинні відповідати бізнес-процеси (регламент виконання процесів, підтримка рівня матеріальних запасів, рівень якості продукції і т. д.).

На концептуальному рівні встановлюються бізнес-правила, що визначають умови виклику функцій у разі виникнення подій і досягнення станів об'єктів.

На внутрішньому рівні виконується формалізація бізнес-правил у вигляді тригерів чи викликів програмних модулів.

4. Організаційна структура.

Організаційна структура становить сукупність взаємозалежних організаційних одиниць, як правило, пов'язаних ієрархічними і процесними відношеннями. **Організаційна одиниця** – це підрозділ, що становить об'єднання людей (персоналу) для виконання сукупності загальних функцій чи бізнес-процесів. У функціонально-орієнтованій організаційній структурі організаційна одиниця виконує набір функцій, які відносяться до однієї функції управління і входять у різні процеси. У процесно-орієнтованій структурі організаційна одиниця виконує набір функцій, що входять в один тип процесу і відносяться до різних функцій управління.

На зовнішньому рівні будується структурна модель підприємства у вигляді ієрархії підпорядкування організаційних одиниць чи списків підрозділів, що взаємодіють.

На концептуальному рівні для кожного підрозділу задається організаційно-штатна структура посад (ролей персоналу).

На внутрішньому рівні визначаються вимоги до прав доступу персоналу до функцій інформаційної системи, що автоматизуються.

5. Технічна структура.

Топологія визначає територіальне розміщення технічних засобів у

структурних підрозділах підприємства, а **комунікація** – технічний спосіб реалізації взаємодії структурних підрозділів.

На зовнішньому рівні моделі визначаються типи технічних засобів обробки даних і їх розміщення у структурних підрозділах.

На концептуальному рівні визначається спосіб комунікацій між технічними комплексами структурних підрозділів: фізичне переміщення документів, машинних носіїв, обмін інформацією каналами зв'язку і т. д.

На внутрішньому рівні будується модель клієнт-серверної архітектури обчислювальної мережі.

Описані моделі предметної області націлені на проектування окремих компонентів ІС: даних, функціональних програмних модулів, управлінських програмних модулів, програмних модулів інтерфейсів користувачів, структури технічного комплексу. Для більш якісного проектування зазначених компонентів потрібна побудова моделей, що погоджують різні моделі між собою. У найпростішому випадку в якості таких моделей взаємодії можуть використовуватися матриці перехресних посилань: "об'єкти – функції", "функції – події", "організаційні одиниці – функції", "організаційні одиниці – об'єкти", "організаційні одиниці – технічні засоби" і т. д. Такі матриці не наочні і не відбивають особливостей реалізації взаємодій.

Для правильного відображення взаємодій компонентів ІС важливо здійснювати спільне моделювання компонентів, що взаємодіють особливо зі змістової точки зору об'єктів і функцій. У цьому плані існують різні методології моделювання предметної області, серед яких варто виділити структурні, об'єктні і комбіновані процесно-орієнтовані підходи.

11.2. Структурний підхід до проектування ІС

11.2.1. Сутність структурного аналізу

В основі структурного аналізу лежить виявлення структури як відносно стійкої сукупності відносин, визнання методологічного переважного значення відносин над елементами в системі, часткове відвернення від розвитку об'єктів. Основним заняттям структурного аналізу є структурний **елемент**, що виконує одну з елементарних функцій, пов'язаних із модельованим предметом, процесом або явищем.

Базовими елементами в процесі використання структурного підходу є модулі. **Модуль** повинен мати наступні властивості:

ім'я, за яким до нього можна звертатися як до єдиного цілого;
приймати і (чи) передавати дані у вигляді параметрів.

Сутність структурного заходу до розробки ІС полягає в її

декомпозиції на функції, що автоматизуються: система розбивається на функціональні підсистеми, які, у свою чергу, діляться на підфункції, що підрозділяються на завдання і т. д. Розбиття продовжується аж до конкретних процедур. При цьому система, що автоматизується, зберігає цілісне подання, в якому всі складові компоненти взаємопов'язані.

Мета структурного аналізу полягає в перетворенні загальних, розпливчатих знань про предметну область в точні моделі, що описують різні підсистеми модельованої організації.

Характерні риси структурного аналізу наступні:

- 1) кількість елементів, виділених на кожному рівні, обмежена (звичайно від 3 до 9, при цьому верхня межа обмежується можливостями людського мозку в сприйнятті певної кількості взаємозалежних об'єктів, а нижня межа вибирається з міркувань здорового глузду);
- 2) виділення на кожному рівні тільки істотних елементів;
- 3) використання чітких формальних правил запису;
- 4) послідовне наближення до кінцевого результату.

Структурний аналіз передбачає дослідження системи за допомогою її графічного модельного подання, яке починається із загального огляду і подальшої деталізації, коли система набуває ієрархічної структури із усе більшою кількістю рівнів.

Структурний аналіз здійснюється з дотриманням наступних **принципів**:

- розбиття системи на частини у вигляді "чорних ящиків";
- ієрархічне упорядкування;
- формалізація.

Перший принцип використання так званих "**чорних ящиків**" для розчленування великих систем на частини дає можливість їх спростити. Перевага використання "чорних ящиків" полягає в тому, що їхньому користувачу не потрібно знати, як вони працюють, – потрібно знати лише їх входи і виходи, а також їх призначення (тобто функції, що вони виконують).

Таким чином, першим кроком спрощення складної системи є її поділ на "чорні ящики" (принцип "поділяй і пануй" – принцип вирішення складних проблем розбиттям їх на множину незалежних завдань, легких для розуміння і вирішення), при цьому цей поділ повинен задовольняти такі критерії:

- а) кожний "чорний ящик" повинен реалізовувати одну-єдину функцію системи;
- б) функція кожного "чорного ящика" повинна бути легко зрозумілою незалежно від складності її реалізації;
- в) зв'язок між "чорними ящиками" повинен вводитися тільки за

наявності зв'язку між відповідними функціями;

г) зв'язки між "чорними ящиками" повинні бути якомога простішими для забезпечення незалежності між ними.

Другий принцип **ієрархічного упорядкування** декларує, що упорядкування виділених частин є важливим для розуміння проблеми. Для розуміння складної системи потрібно виділені частини організувати як ієрархічні структури. Система може бути зрозумілою, коли вона розбудована за рівнями, кожний із яких додає нових деталей.

Третій принцип – **формалізації** – полягає в необхідності використання графічних нотацій, які відображають структуру системи, елементи даних, етапи обробки за допомогою спеціальних графічних символів діаграм, а також описують проект системи формальними і природними мовами. Нотації полегшують розуміння складних систем. Відомо, що "одна картинка коштує тисячі слів".

11.2.2. Методології структурного підходу

Методології структурного аналізу традиційно використовують моделі, що показують:

- функції, які система повинна виконувати;
- процеси, що забезпечують виконання функцій;
- дані, що необхідні під час виконання функцій, і відношення між ними;
- організаційні структури, що забезпечують виконання функцій;
- матеріальні та інформаційні потоки, що виникають у ході виконання функцій.

Для побудови перерахованих моделей у методологіях структурного аналізу найбільш часто застосовуються такі діаграми:

1) діаграми функціональних специфікацій, діаграми потоків даних, діаграми переходів станів у нотаціях:

SADT (Structured Analysis and Design Technique);

Йордана (Yourdon);

Гейна – Сарсона (Gane – Sarson);

SAG (Software AG);

2) діаграми моделей даних "сутність – зв'язок" у нотаціях:

Чена (Chen);

Беркера;

SADT;

SAG;

3) діаграми структури програмного додатку в нотаціях:

Константайна (Constantine);

Джексона (Jackson);

Варньє – Орра (Warnier – Orr);

SAG.

У якості інструментальних засобів структурного аналізу і проектування найбільш часто і ефективно застосовуються наступні:

BFD (Business Function Diagrams) – діаграми бізнес-функцій (функціональні специфікації);

DFD (Data Flow Diagrams) – діаграми потоків даних у нотаціях Гейна – Сарсона, Йордана та інших, що забезпечують аналіз і функціональне проектування інформаційних систем;

ERD (Entity-Relationship Diagrams) – діаграми "сутність – зв'язок" у нотаціях Чена и Баркера;

STD (State Transition Diagrams) – діаграми переходів станів, які засновані на розширеннях Уорда – Меллора (Ward – Mellor) і Хатлі (Hatley) для проектування систем реального часу;

SSD (System Structure Diagrams) – діаграми структури програмного додатка.

FDD (Functional Decomposition Diagrams) – діаграми функціональної декомпозиції;

сімейство **IDEF (Integration Definition for Function Modeling)**:

IDEF0 – методологія функціонального моделювання, що є складовою частиною SADT і що дозволяє описати бізнес-процес у вигляді ієрархічної системи взаємопов'язаних функцій;

IDEF1 – методологія аналізу і вивчення взаємозв'язків між інформаційними потоками в рамках комерційної діяльності підприємства;

IDEF1X – методологія інформаційного моделювання, заснована на концепції "сутність – зв'язок" Ченом. Застосовується для розробки реляційних баз даних і використовує умовний синтаксис, що спеціально розроблений для зручної побудови концептуальної схеми і забезпечує універсальне подання структури даних в рамках підприємства, незалежне від кінцевої реалізації бази даних і апаратної платформи;

IDEF3 – методологія моделювання потоків робіт підприємства, що дозволяє подати їх сценарії за допомогою опису послідовності змін властивостей об'єкта в рамках даного процесу;

IDEF4 – методологія об'єктно-орієнтованого проектування для підтримки проектів, пов'язаних з об'єктно-орієнтованими реалізаціями;

IDEF5 – методологія, що забезпечує наочне подання даних, отриманих у результаті обробки онтологічних запитів, у простій, графічній формі.

Усі перелічені засоби містять графічні і текстові засоби моделювання: перші – для зручності відображення основних компонентів моделі, другі – для забезпечення точного визначення її компонентів і зв'язків.

Методології структурного аналізу і проектування можна класифікувати наступним чином (рис. 41).

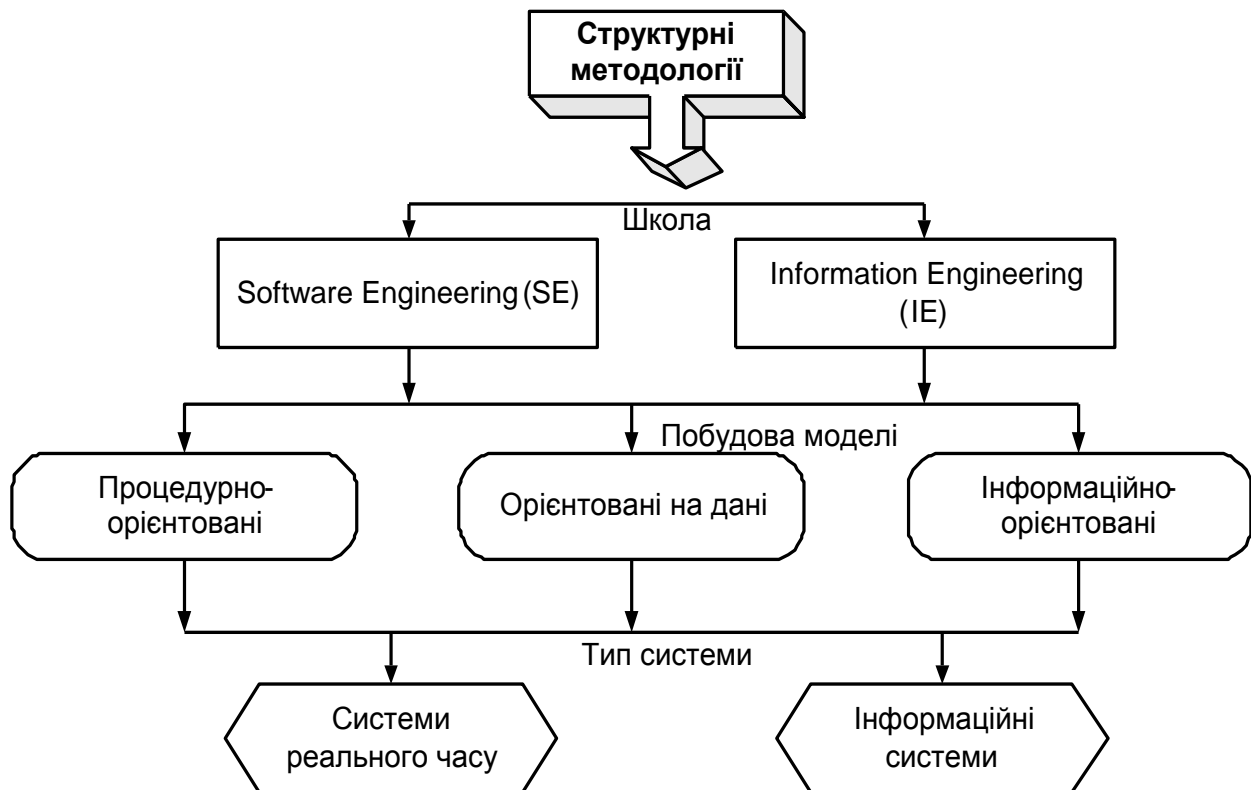


Рис. 41. Класифікація структурних методологій

За відношенням до шкіл виділяють методології програмної інженерії (Software Engineering, SE) та інформаційної інженерії (Information Engineering, IE).

SE є нисхідним поетапним підходом до розробки ПЗ, що починається із загального погляду на його функціонування. Потім проводиться декомпозиція на підфункції. Процес повторюється доти, поки підфункції не стануть достатньо малими для можливості їх кодування. У результаті виходить ієрархічна, структурована модульна програма. SE подає універсальні методології розробки ПЗ.

IE представляє нові методології побудови систем взагалі і включає етапи більш високого рівня (наприклад, стратегічне планування). На етапі проектування систем за методологією SE і IE аналогічні.

Залежно від порядку побудови моделі виділяють методології процедурно-орієнтовані, орієнтовані на дані та інформаційно-орієнтовані. Процедурно-орієнтований підхід регламентує первинність проектування функцій порівняно з даними: вимоги до даних

розкриваються через функціональні вимоги. Підхід орієнтований на дані найбільш важливими є вхід і вихід системи, тому структури даних визначаються в першу чергу, а процедурні компоненти є похідними від даних. Третій, інформаційно-орієнтований підхід, є складовою ІЕ-дисципліни, його відмінна риса від підходу, орієнтованого на дані, – можливість працювати з ієрархічними структурами даних.

Залежно від **типу проєктованих систем** виділяють методології для систем реального часу та методології для інформаційних систем. Основна відмінність цих типів систем полягає в наступному: системи реального часу контролюють і контролюються зовнішніми подіями, а інформаційні системи – даними.

Найбільш істотна відмінність між різновидами структурного аналізу і проєктування полягає у використуваних методах і засобах функціонального моделювання. З цього погляду всі різновиди структурного аналізу можуть бути розділені на дві групи: у якій застосовуються методи і технологія DFD (у різних нотаціях) і ті, що використовують SADT-методологію.

За допомогою цих методологій можуть бути побудовані логічні моделі предметної сфери: початкової – **як є (as is)** і реорганізованої – **як повинно бути (to be)**.

Далі слід розглянути більш детально деякі з найбільш поширених методологій структурного аналізу і проєктування.

11.2.3. Методологія функціонального моделювання SADT

SADT – одна з найвідоміших технологій структурного аналізу і проєктування. Розроблена вона в 1973 р. Росом (Ross). Стандарт **IDEF0** є підмножиною SADT і підтримується CASE-засобами BPWin. Згідно з методологією моделі розробляються на основі подання функцій ІС або її елементів (планів, даних, устаткування і т. д.). Відповідні моделі прийнято називати активнісними моделями і моделями даних. Активнісна модель становить сукупність активностей, взаємопов'язаних через об'єкти (елементи) системи. Моделі даних є описом об'єктів, пов'язаних системними активностями.

Методологія SADT є сукупністю методів, правил і процедур, призначених для побудови функціональної моделі об'єкта якої-небудь предметної області. Функціональна модель SADT відображає функціональну структуру об'єкта, тобто виконувані ним дії і зв'язки між цими діями. Основні елементи цієї методології ґрунтуються на наступних концепціях:

графічного подання блочного моделювання (табл. 15). Графіка блоків і дуг SADT-діаграми відображає функцію у вигляді блоку, а інтерфейси входу/виходу зображуються дугами, що відповідно входять у блок і виходять з нього. Взаємодія блоків один з одним описується за допомогою інтерфейсних дуг, що виражають "обмеження", які, у свою чергу, визначають, коли і яким чином функції виконуються і управляються;

чіткості й точності. Виконання правил SADT вимагає достатньої строгості й точності, не накладаючи водночас надмірних обмежень на дії аналітика. Правила SADT включають:

обмеження кількості блоків на кожному рівні декомпозиції (як правило, 3 – 6 блоків);

зв'язності діаграм (номерів блоків);

унікальності міток і найменувань (відсутності імен, що повторюються);

синтаксичних правил для графіки (блоків і дуг);

поділу входів і управлінь (правило визначення ролі даних).

Методологія SADT може використовуватися для моделювання широкого кола систем і визначення вимог і функцій, а потім для розробки системи, яка відповідає цим вимогам і реалізує ці функції. Для вже наявних систем SADT може бути використана з метою аналізу функцій, що виконуються системою, а також зазначення механізмів, за допомогою яких вони здійснюються.

Моделі ІС згідно з методологією подаються у вигляді діаграм, які ієрархічно впорядковані. Активності на діаграмах зображуються прямокутниками і супроводжуються описами природною мовою. Дуги на діаграмах становлять набори предметів і також супроводжуються пояснювальними текстами. Вхідні дуги відображають предмети, використовувані і перетворювані активностями. Керівні дуги відображають управління діями активностей. Вихідні дуги відображають предмети, які є реалізацією активності.

Об'єкти, використовувані в нотації IDEF0

№	Найменування об'єкта	Опис	Графічне подання
1	Робота (Activity)	Слугує для опису функцій (процедур, завдань), що виконуються підрозділами (співробітниками) підприємства протягом певного часу і що мають певні результати	
2	Вхід (Input)	Описує вхідні документи, інформацію, матеріальні ресурси, необхідні для виконання функції	
3	Вихід (Output)	Описує документи, що виходять, інформацію, матеріальні ресурси, виконання функції, що є результатом	
4	Управління (Control)	Описує дії, що управляють: обмеження, закони, нормативні документи і т. д.	
5	Механізм (Mechanism)	Описує механізми, тобто ресурси, що необхідні для виконання функції і не змінюють свого стану в процесі виконання функції	
6	Стрілка передування	З'єднує послідовно виконувани функції	
7	Стрілка відношення	Використовується для узгодження коментарів з функціями	
8	Стрілка потоку об'єктів	Показує потік об'єктів від однієї функції до іншої	

Активності на діаграмі розміщуються відповідно до їх домінування по діагоналі робочого документа. На діаграмах можна відбивати лише 5 типів взаємозв'язків між активностями:

вихід – управління;

вихід – вхід;

вихід – управління за зворотним зв'язком;

вихід – вхід за зворотним зв'язком;

вихід – механізм виконання.

Дуги на діаграмах можуть розгалужуватися і з'єднуватися. Розгалуження дуги означає, що частина її вмісту або весь набір об'єктів може з'явитися в кожній гілці. Дуги позначаються текстом до гілки і після. Текстова мітка уточнює, що містить кожна частина дуги. У разі злиття дуг текстова мітка може бути встановлена на кожній гілці. Після вузла злиття текстова мітка повинна бути проставлена обов'язково для зазначення нового набору об'єктів.

Результатом застосування методології SADT є модель, яка складається з діаграм, фрагментів текстів і глосарію, що мають посилання один на одного. Діаграми – головні компоненти моделі, всі функції ІС та інтерфейси на них подані як блоки і дуги. Місце з'єднання дуги з блоком визначає тип інтерфейсу. Керівна інформація входить у блок зверху, тоді як інформація, яка піддається обробці, показана з лівого боку блоку, а результати виходу показані з правого боку. Механізм (людина або автоматизована система), який здійснює операцію, зображується дугою, що входить в блок знизу (рис. 42).

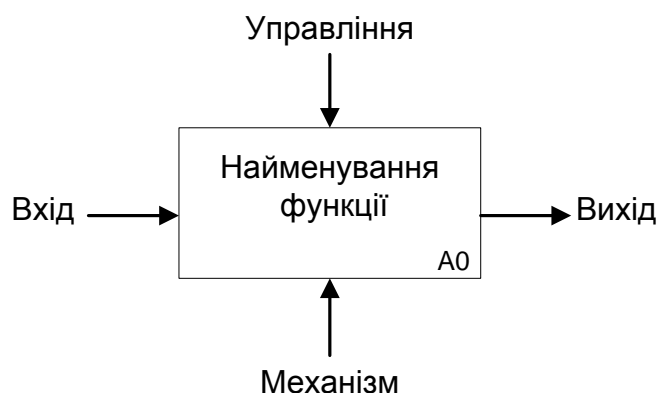


Рис. 42. Функціональний блок та інтерфейсні дуги

Однією з найбільш важливих особливостей методології SADT є поступове введення у міру створення діаграм все більших рівнів деталізації. Кожен компонент моделі може бути декомпозований на іншій діаграмі. Кожна діаграма ілюструє "внутрішню будову" блоку на батьківській діаграмі.

Побудова SADT-моделі починається з подання всієї системи у вигляді простої компоненти – одного блоку і дуг, що зображують інтерфейси з функціями поза системою. Оскільки єдиний блок подає всю систему як єдине ціле, ім'я, вказане в блоці, є спільним. Це правильно й для інтерфейсних дуг – вони також становлять повний набір зовнішніх інтерфейсів системи в цілому.

Потім блок, який подає систему як єдиний модуль, деталізується на іншій діаграмі за допомогою декількох блоків, з'єднаних інтерфейсними дугами. Ці блоки становлять основні підфункції початкової функції. Дана декомпозиція виявляє повний набір підфункцій, кожна з яких представлена як блок, межі якого визначені інтерфейсними дугами. Кожна з цих підфункцій може бути декомпозована так само для детальнішого подання.

У всіх випадках кожна підфункція може містити лише ті елементи, які входять в початкову функцію. Крім того, модель не може опустити які-небудь елементи, тобто, як уже наголошувалося, батьківський блок і його інтерфейси забезпечують контекст. До нього не можна нічого додати, і з нього не може бути нічого видалено.

Модель SADT є серією діаграм із супровідною документацією, що розбивають складний об'єкт на складові частини, які подані у вигляді блоків. Деталі кожного з основних блоків показані у вигляді блоків на інших діаграмах. Кожна детальна діаграма є декомпозицією блоку з більш загальної діаграми. На кожному кроці декомпозиції більш загальна діаграма називається батьківською для детальнішої діаграми.

Дуги, що входять в блок і виходять з нього на діаграмі верхнього рівня, є такими самими, що й дуги, які входять в діаграму нижнього рівня і виходять з неї, тому що блок і діаграма становлять одну і ту ж роботу.

Деякі дуги приєднані до блоків діаграми обома кінцями, в інших же один кінець залишається неприєднаним. Неприєднані дуги відповідають входам, управлінням і виходам батьківського блоку. Джерело або одержувач цих дуг може бути виявлений лише на батьківській діаграмі. Неприєднані кінці повинні відповідати дугам на початковій діаграмі. Усі граничні дуги повинні продовжуватися на батьківській діаграмі, щоб вона була цілісною і несуперечливою.

На SADT-діаграмах явно не вказана ні послідовність, ні час. Зворотні зв'язки, ітерації, процеси, що тривають, і функції, що перекриваються (за часом), можуть бути зображені за допомогою дуг. Зворотні зв'язки можуть виступати у вигляді коментарів, зауважень, виправлень і т. д.

Як було зазначено, механізми (дуги з нижнього боку) показують засоби, за допомогою яких здійснюється виконання функцій. Механізм

може бути людиною, комп'ютером або будь-яким іншим пристроєм, який допомагає виконувати дану функцію.

Кожен блок на діаграмі має свій номер. Блок будь-якої діаграми може бути далі описаний діаграмою нижнього рівня, яка, у свою чергу, може бути далі деталізована за допомогою необхідної кількості діаграм. Таким чином формується ієрархія діаграм.

11.2.4. Методології моделювання потоків даних

Обидві методології засновано на потоках даних. Їх головне призначення полягає у створенні заснованих на графіці документів щодо функціональних вимог. Методології підтримуються традиційними нисхідними методами проектування специфікацій і забезпечують один із найкращих способів зв'язку між аналітиками, розробниками і користувачами системи за рахунок інтеграції наступних засобів:

діаграм потоків даних DFD;

словників даних, включаючи групові й індивідуальні потоки даних, сховища і процеси, всі атрибути;

мініспецифікацій обробки, що описують процеси нижнього рівня, який є базою кодогенерації.

Відповідно до методології модель системи визначається як ієрархія діаграм потоків даних, що описують асинхронний процес перетворення інформації від її введення в систему до видачі користувачеві. 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 (зовнішня сутність) 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, накопичувачів даних або зовнішньої сутності – споживачів інформації. Таким чином, основними компонентами діаграм потоків даних є (табл. 16):

- 1) зовнішня сутність;
- 2) робота;
- 3) сховище даних;
- 4) потік даних.

Першим кроком у процеси побудови ієрархії DFD є побудова контекстних діаграм. Зазвичай під час проектування відносно простих ІС будується єдина контекстна діаграма із зіркоподібною топологією, в

центрі якої перебуває так званий головний процес, поєднаний із приймачами із джерелами інформації, за допомогою яких із системою взаємодіють користувачі та інші зовнішні системи.

Якщо ж для складної системи обмежитися єдиною контекстною діаграмою, то вона міститиме занадто велику кількість джерел і приймачів інформації, які важко розташувати на аркуші 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 (десяти й більше);

розподілена природа системи;

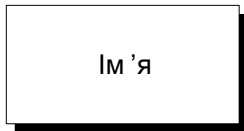
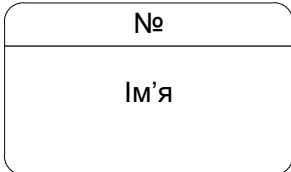
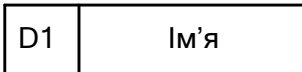
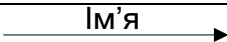
багатофункціональність системи із групуванням функцій, що вже склалося або виявлено, в окремі підсистеми.

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

Ієрархія контекстних діаграм визначає взаємодію основних функціональних підсистем проектованою ІС як між собою, так і із зовнішніми вхідними і вихідними потоками даних і зовнішніми об'єктами (джерелами і приймачами інформації), з якими взаємодіє ІС.

Таблиця 16

Об'єкти, використовувані в нотації DFD

Найменування об'єкта	Опис	Графічне подання (Гейна – Сарсона)
Зовнішня сутність	є матеріальним предметом або фізичною особою, що є джерелом або приймачем інформації, наприклад, замовники, персонал, постачальники, клієнти, склад	
Робота	є функціями системи щодо перетворення вхідних потоків даних у вихідні відповідно до певного алгоритму. У верхній частині зазначається номер роботи, який слугує для її ідентифікації. У поле імені вводиться найменування роботи	
Сховище даних	є абстрактним пристроєм для зберігання інформації, яку можна у будь-який момент помістити у сховище і через деякий час витягувати. Сховище даних ідентифікується літерою "D" і довільним числовим позначенням	
Потік даних	визначає інформацію, що передається через певне сполучення від джерела до приймача. Реальний потік даних може бути інформацією, що передається за допомогою кабеля між двома пристроями, листами, що пересилаються поштою, і т. д. Кожен потік даних має ім'я, зміст, що відбиває його	

Розробка контекстних діаграм вирішує проблему чіткого визначення функціональної структури ІС на більш ранній стадії її проектування, що особливо важливо для складних багатофункціональних систем, у розробці яких беруть участь різні організації і колективи розробників.

Після побудови контекстних діаграм отриману модель слід перевірити на повноту початкових даних про об'єкти системи й ізольованість об'єктів (відсутність інформаційних зв'язків з іншими об'єктами).

Для кожної підсистеми, присутньої на контекстних діаграмах, виконується її деталізація за допомогою DFD. Кожен процес на DFD, у свою чергу, може бути деталізований за допомогою DFD або мініспецифікацій. У ході деталізації повинні виконуватися наступні правила:

правило балансування – означає, що під час деталізації підсистеми або процесу діаграма, яка деталізує зовнішні джерела/приймачі даних може мати лише ті компоненти (підсистеми, процеси, зовнішні сутності, накопичувачі даних), з якими має інформаційний зв'язок підсистема, що деталізується, або процес на батьківській діаграмі;

правило нумерації – означає, що в ході деталізації процесів повинна підтримуватися їх ієрархічна нумерація. Наприклад, процеси, що деталізують процес з номером 2, отримують номери 2.1, 2.2, 2.3 і т. д.

Мініспецифікація (опис логіки процесу) повинна формулювати основні функції так, щоб надалі фахівець, що виконує реалізацію проекту, зміг виконати їх або розробити відповідну програму.

Мініспецифікація є кінцевою вершиною ієрархії DFD. Рішення про завершення деталізації процесу і використання мініспецифікації ухвалюється аналітиком на основі таких критеріїв:

наявності в 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 обсягу (не більше 20 – 30 рядків).

У процесі побудови ієрархії DFD переходити до деталізації процесів

слід лише після визначення змісту всіх потоків і накопичувачів даних, яке описується за допомогою структур даних. Структури даних конструюються з елементів даних і можуть містити альтернативи, умовні входження та ітерації. Умовне входження означає, що даний компонент може бути відсутнім у структурі. Альтернатива означає, що в структуру може входити один із перерахованих елементів. Ітерація означає входження будь-якої кількості елементів у вказаному діапазоні. Для кожного елементу даних може зазначатися його тип (безперервні або дискретні дані). Для безперервних даних може зазначатися одиниця вимірювання (кг, см і т. д.), діапазон значень, точність передачі 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 й деталізовані. Виявлені недеталізовані об'єкти слід деталізувати, повернувшись на попередні кроки розробки. В узгодженій моделі для всіх потоків даних і накопичувачів даних повинне виконуватися правило збереження інформації: всі дані, що куди-небудь надходять, повинні бути зчитані, а всі зчитувані дані повинні бути записані.

11.2.5. Методології моделювання даних

Метод Беркера. Мета моделювання даних полягає в забезпеченні розробника ІС концептуальною схемою бази даних у формі однієї моделі або декількох локальних моделей, які відносно легко можуть бути відображені в будь-якій системі баз даних.

Найбільш поширеним засобом моделювання даних є діаграми "сутність – зв'язок" (ERD). З їх допомогою визначаються важливі для предметної області об'єкти (сутності), їх властивості (атрибути) і відносини один з одним (зв'язки). ERD безпосередньо використовуються для проектування реляційних баз даних. Нотація ERD була вперше запроваджена П. Ченом (Chen) і отримала подальший розвиток у роботах Беркера.

Методологія IDEF1. Метод IDEF1, розроблений Т. Ремей (T. Ramey), також заснований на підході П. Чена і дозволяє забудувати модель даних, еквівалентну реляційній моделі в третій нормальній формі. Зараз

на основі вдосконалення методології IDEF1 створена її нова версія – методологія IDEF1X. IDEF1X розроблена з урахуванням таких вимог, як простота вивчення і можливість автоматизації. IDEF1X-діаграми використовуються разом поширених CASE-засобів (зокрема, ERWin, Design/IDEF).

11.2.6. Методологія моделювання потоків робіт IDEF3

Нотація IDEF3, яка називається ще **workflow diagramming** (діаграма потоків робіт) використовується для опису логіки взаємодії інформаційних потоків у ході виконання певних процесів. Ця методологія призначена для моделювання взаємин між процесами обробки інформації і об'єктів, що є частиною цих процесів. Діаграма потоків робіт дозволяє:

описати виконання в певній послідовності процесів разом з об'єктами;

провести аналіз завершеності процедур обробки інформації;

описати сценарії дій співробітників організації.

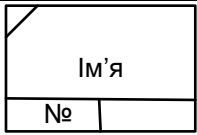
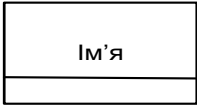
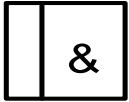
Для побудови моделі в нотації IDEF3 використовуються об'єкти, характеристика яких наведена в табл. 17.


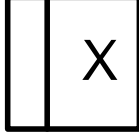
У моделі використовуються такі ж види стрілок, як в IDEF0 (див. табл. 15).

У ході побудови моделі зв'язки задають відношення між роботами. Усі зв'язки мають один напрямок, можуть бути спрямовані куди завгодно. Однак існує правило, що рекомендує спрямовувати стрілки зліва направо.

Таблиця 17

Об'єкти, використовувані в нотації IDEF3

№	Найменування	Опис	Графічне подання
1	2	3	4
1	Робота (unit of work activity)	Слугує для опису функцій (процедур, завдань), що виконуються підрозділами (співробітниками) підприємства	
2	Посилальний об'єкт	Об'єкт, використовуваний для опису посилань на інші діаграми моделі, циклічні переходи в рамках однієї моделі, коментарі до функцій	
3	Логічне "І"	Логічний оператор, що визначає зв'язки між функціями. У разі злиття, всі попередні процеси повинні бути завершені. У процесі розгалуження всі наступні процеси повинні бути запущені	

1	2	3	4
4	Логічне "АБО"	Логічний оператор, що визначає зв'язки між функціями. У разі злиття один або декілька попередніх процесів повинні бути завершені. У разі розгалуження один або декілька таких процесів повинні бути запуснені	
5	Що виключає "АБО"	Логічний оператор, що визначає зв'язки між функціями. У разі злиття лише один з попередніх процесів повинен бути завершений. У разі розгалуження лише один із таких процесів повинен бути запуснений	

11.2.7. Структурне проектування

У процесі структурного проектування виконуються два види робіт:

1) проектування архітектури ІС, що включає:

розробку структури й інтерфейсів її компонентів (автоматизованих робочих місць);

узгодження функцій і технічних вимог до компонентів;

визначення інформаційних потоків між основними компонентами, зв'язків між ними і зовнішніми об'єктами;

2) детальне проектування, що включає:

розробку специфікацій кожного компонента;

розробку вимог до тестів і плану інтеграції компонентів;

побудову моделей ієрархії програмних модулів і міжмодульних взаємодій, проектування внутрішньої структури модулів.

При цьому відбувається розширення моделі вимог за рахунок:

уточнення наявних функціональних, інформаційних і, можливо, подієвих моделей вимог, побудованих за допомогою відповідних засобів структурного аналізу;

побудови моделей автоматизованих робочих місць, що включають інформаційну модель і орієнтовані на неї функціональні моделі (аж до ідентифікації конкретних сутностей інформаційної моделі);

побудови моделей міжмодульних і внутрішньомодульних взаємодій з використанням техніки структурних карт.

У структурному підході з метою проектування модулів використовується техніка **структурних карт** (схем), що демонструє, яким чином системні вимоги будуть відбиватися комбінацією програмних структур. При цьому найчастіше застосовують два види структурних карт:

структурні карти Константайна, призначені для опису відношень між модулями;

структурні карти Джексона – для опису внутрішньої структури модулів.

Структурні карти Константайна є моделлю відносин ієрархії між

програмними модулями. Вузли структурних карт відповідають модулям і областям даних, потоки відображають міжмодульні виклики (у тому числі циклічні, умовні й рівнобіжні). Міжмодульні зв'язки за даними і управлінням також моделюються спеціальними вузлами, прив'язаними до потоків, стрілками вказуються напрямки потоків і зв'язків. Фундаментальні елементи структурних карт Константайна стандартизовані IBM, ISO і ANSI.

Техніка структурних карт Джексона зводиться до методології структурного програмування Джексона і полягає в проектуванні діаграм і схем для графічного ілюстрування внутрішньомодульних (а іноді й міжмодульних) зв'язків і документування проекту архітектури ІС. При цьому структурні карти Джексона дозволяють здійснювати проектування нижнього рівня ІС і на цьому етапі є близькими до традиційних блок-схем, що моделюють послідовне, рівнобіжне, умовне й ітераційне виконання їх вузлів.

11.3. Об'єктний підхід до проектування ІС

11.3.1. Сутність об'єктного підходу

Об'єктно-орієнтовані методології засновані на об'єктній декомпозиції предметної області. Предметна область подається у вигляді сукупності об'єктів, що взаємодіють між собою за допомогою передачі повідомлень. В об'єктній методології використовуються поняття, визначення яких наведено в табл. 18.

Як об'єкти предметної області можуть розглядатися конкретні предмети, а також абстрактні чи реальні сутності (наприклад, клієнт, замовлення, підприємство тощо). Кожен об'єкт характеризується своїм станом (точніше, набором атрибутів, значення яких визначають стан), а також набором операцій для перевірки і зміни цього стану.

Таблиця 18

Визначення термінів об'єктно-орієнтованої методології

Термін	Англійський варіант	Значення терміна
1	2	3
ідентичність	identity	Природа об'єкта: те, що відрізняє його від інших об'єктів
клас	class	Множина об'єктів зі спільною структурою і поведінкою. Терміни "клас" і "тип" взаємозамінні

1	2	3
об'єктно-орієнтована декомпозиція	object-oriented decomposition	Процес розбиття системи на частини, відповідні класам і об'єктам предметної області. Практичне застосування методів об'єктно-орієнтованого проектування приводить до об'єктно-орієнтованої декомпозиції, за якої ми розглядаємо світ як сукупність об'єктів, що узгоджено діють для забезпечення необхідної поведінки
об'єктно-орієнтоване програмування	object-oriented programming (OOP)	Методологія реалізації, за якої програма організовується як сукупність об'єктів, що співробітничать, кожен із яких є екземпляром якого-небудь класу, а класи утворюють ієрархію спадкоємства. При цьому класи зазвичай статичні, а об'єкти дуже динамічні, що заохочується динамічним зв'язуванням і поліморфізмом
об'єктно-орієнтоване проектування	object-oriented design (OOD)	Методологія проектування, що поєднує процес об'єктно-орієнтованої декомпозиції і систему позначень для подання логічної і фізичної, статичної та динамічної моделей проектованої системи. Система позначень складається з діаграм класів, об'єктів, модулів і процесів
об'єктно-орієнтований аналіз	object-oriented analysis	Метод аналізу, згідно з яким вимоги розглядаються з погляду класів і об'єктів, що є складовими словника предметної області
операція	operation	Дія, що виконується одним об'єктом над іншим, щоб викликати реакцію. Усі операції, які можна виконати над яким-небудь об'єктом, зосереджені у вільних підпрограмах і функціях-членах (методах). Терміни "операція" і "метод" взаємозамінні
поведінка	behavior	Дії та реакції об'єкта, виражені в термінах передачі повідомлень і зміни стану; видима ззовні і відтворена активність об'єкта
стан	state	Сукупний результат поведінки об'єкта: одна зі стабільних умов, у яких об'єкт може існувати, охарактеризованих кількісно; у будь-який конкретний момент часу стан об'єкта включає перелік (зазвичай, статичний) властивостей об'єкта і поточні значення (зазвичай, динамічні) цих властивостей

Кожен об'єкт є представником певного класу однотипних об'єктів, що визначає його загальні властивості. Усі представники (екземпляри) того самого класу мають той самий набір операцій і можуть реагувати на ті самі повідомлення.

Об'єкти і класи організуються з дотриманням таких **принципів**:

1. Принцип **інкапсуляції** (приховання інформації) декларує заборону будь-якого доступу до атрибутів об'єкта, крім одного – операції. Відповідно до цього внутрішня структура об'єкта прихована від користувача, а будь-яка його дія ініціюється зовнішнім повідомленням, що обумовлює виконання відповідної операції.

2. Принцип **спадкування** декларує створення нових класів – від загального до окремого. Такі нові класи зберігають усі властивості класів-батьків і при цьому містять додаткові атрибути й операції, що характеризують їхню специфіку.

3. Принцип **поліморфізму** декларує можливість роботи з об'єктом

без інформації про конкретний клас, представником якого він є. Кожен об'єкт може вибирати операцію на основі типів даних, прийнятих у повідомленні, тобто реагувати індивідуально на це повідомлення (те саме для різних об'єктів).

Таким чином, об'єктно-орієнтований підхід полягає в поданні системи, що моделюється, у вигляді сукупності класів і об'єктів предметної сфери. При цьому ієрархічний характер складної системи виявляється з використанням ієрархії класів, а її функціонування розглядається як взаємодія об'єктів.

Об'єктно-орієнтовані методології базуються на інтегрованих моделях трьох типів:

об'єктної моделі, що відображає ієрархію класів, пов'язаних спільністю структури і поводження і які відображають специфіку атрибутів і операцій кожного з них;

динамічної моделі, що відображає часові аспекти й послідовність операцій;

функціональної моделі, що описує потоки даних.

Якщо методи структурного проектування мали на меті спрощення системної розробки на основі алгоритмічного підходу, то об'єктно-орієнтовані методи вирішують аналогічне завдання, використовуючи опис класів і об'єктів, тобто чіткі засоби об'єктно-орієнтованого проектування. М. Буч визначив об'єктно-орієнтоване проектування як "методологію проектування, що поєднує в собі процес об'єктної декомпозиції і прийоми подання як логічної і фізичної, так і статичної та динамічної моделей проектованої системи".

Кінцевим результатом процесу об'єктно-орієнтованого проектування повинна стати множина класів об'єктів із приєднаними методами обробки атрибутів. Якщо в структурному підході моделі даних і операцій розробляються незалежно одна від одної і тільки координуються між собою, то об'єктно-орієнтований підхід передбачає спільне моделювання даних і процесів. При цьому моделі проблемної сфери застосування уточнюються.

Система об'єктно-орієнтованих моделей послідовно розвертається в напрямку від статичної моделі загального подання функціонування ІС до моделі динамічної взаємодії об'єктів, на основі якої можуть бути згенеровані класи об'єктів у конкретному програмно-технічному середовищі.

11.3.2. Стандарти об'єктного проектування

На сьогодні для об'єктно-орієнтованого моделювання проблемної сфери широко використовується **уніфікована мова моделювання UML** (Unified Modeling Language), яка розроблена групою провідних комп'ютерних фірм світу OMG (Object Management Group) і фактично є стандартом з об'єктно-орієнтованих технологій. Мова UML реалізована багатьма фірмами-виробниками програмного забезпечення в рамках CASE-технологій, наприклад Rational Rose (Rational), Natural Engineering Workbench (Software AG), ARIS Toolset (IDS prof. Scheer) та ін.

Уніфікована мова моделювання є графічною мовою для візуалізації, специфікації, конструювання і документування систем, у яких велика роль належить програмному забезпеченню. За допомогою мови UML можна розробити детальний проект створюваної системи, що відображає:

концептуальні елементи: системні функції і бізнес-процеси;

конкретні особливості реалізації: класи, написані спеціальними мовами програмування, схеми баз даних, програмні компоненти багаторазового використання.

Система об'єктно-орієнтованих моделей відповідно до нотацій UML включає наступні діаграми:

1) діаграму варіантів використання (Use-case diagram), що відображає функціональність ІС у вигляді сукупності послідовностей, що виконуються, транзакцій;

2) діаграму класів об'єктів (Class diagram), що відображає структуру сукупності взаємозалежних класів об'єктів аналогічно до ER-діаграми функціонально-орієнтованого підходу;

3) діаграми станів (State chart diagrams), кожна з яких відображає динаміку станів об'єктів одного класу і пов'язаних з ними подій;

4) діаграми взаємодії об'єктів (Interaction diagrams), кожна з яких відображає динамічну взаємодію об'єктів у рамках одного прецеденту використання;

5) діаграми діяльності (Activity diagrams), що відображають потоки робіт у взаємозалежних прецедентах використання (можуть декомпонуватися на більш детальні діаграми);

6) діаграми пакетів (Package diagrams), що відображають розподіл об'єктів за функціональними чи забезпечувальними підсистемами (можуть декомпонуватися на більш детальні діаграми);

7) діаграму компонентів (Component diagram), що відображає фізичні

модулі програмного коду;


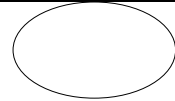

8) діаграму розміщення (Deployment diagram), що відображає розподіл об'єктів між вузлами обчислювальної мережі.

Діаграма варіантів використання

Діаграма варіантів використання виявляє основні бізнес-процеси як послідовності транзакцій, які повинні виконуватися повністю, коли виконання відособленої підмножини дій не має значення без виконання всієї послідовності. Варіанти використання ініціюються із зовнішнього середовища користувачами ІС, так званими **акторами**. На цьому рівні моделювання не розкривається механізм реалізації процесів. Наведені сутності мають графічні позначення, наведені в табл. 19.

Таблиця 19

Об'єкти, використовувані в діаграмі варіантів використання

Найменування об'єкта	Опис	Графічне позначення
Актор	Зовнішній користувач процесу	
Варіант використання	Бізнес-процес	
Пойменована стрілка	Позначення події	

Актор ініціює виконання варіанта використання і отримує від нього результати. Взаємодія (асоціація) актора з варіантом використання здійснюється внаслідок події, яка позначається поймаючою стрілкою. Один актор може брати участь у декількох варіантах використання, а в одному варіанті використання може бути зайнято декілька акторів.

У реалізації варіанта використання можливе виділення декількох потоків подій:

основний потік подій, який приводить до необхідного результату найбільш коротким шляхом;

альтернативні потоки подій.

Основний і альтернативний потоки подій у моделі варіантів використання описуються у вигляді неформальних текстових коментарів.

Декілька варіантів використання можуть мати спільну частину, що виділяється в самостійний варіант використання, з яким установлюються відносини використання (uses). З іншого боку, певні варіанти використання можуть бути розширені деталями. У такому разі створюється додатковий варіант використання, з яким встановлюються відносини розширення (extends).

Діаграми класів об'єктів

Діаграми класів об'єктів відображають статичну структуру класів об'єктів. Ця діаграма розглядає внутрішню структуру предметної сфери, ієрархію класів об'єктів, статичні зв'язки об'єктів.

Класи об'єктів можуть мати різні стереотипи поведінки: об'єкти-сутності, керівні об'єкти, інтерфейсні об'єкти (табл. 20). Об'єкти, відображені в діаграмі класів об'єктів, пов'язуються статичними відносинами, які відбивають постійні зв'язки між об'єктами незалежно від виконання конкретного бізнес-процесу. До статичних відносин відносяться узагальнення, агрегація, асоціація об'єктів.


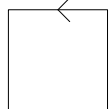

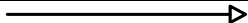

Діаграми станів

Діаграми станів відображають поведінку об'єктів одного класу в динаміці, зв'язок станів об'єктів з подіями і визначає:

- які типові стани проходить об'єкт;
- які події ведуть до зміни стану об'єкта;
- які дії об'єкт виконує, коли він отримує повідомлення про зміну стану;
- як об'єкти створюються і знищуються (вхідні і вихідні точки діаграми).

Таблиця 20

Об'єкти, використовувані в діаграмі класів об'єктів

Найменування об'єкта	Опис	Графічне позначення
Об'єкт-сутність	Пасивний об'єкт, над яким виконуються операції обробки процесу	
Керівний об'єкт	Активний об'єкт, що координує виконання функцій	
Відносини асоціації	Відносини типу 0.. 1:1;0..1:M; M:N. Відносини можуть бути поійменованими. 0...1 – необов'язковість зв'язку; * – множинність зв'язку	
Відносини узагальнення	Відносини спадкоємства	
Відносини агрегації	Відносини "ціле – частина"	

У табл. 21 наведені використовувані в діаграмі станів поняття і їх графічні позначення.

З кожним станом пов'язана одна або більше подій, які можуть його змінити. Для стану задаються імена всіх пов'язаних з ним переходів в інші стани.

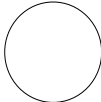


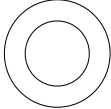
Перехід станів описується наступними атрибутами.

Призначення – стан об'єкта, в який перейде об'єкт після переходу стану.

Виклик – ім'я події, яка спричиняє перехід станів. Імена подій повинні бути ідентичними у визначенні класу і стану. Події, що спричиняють перехід, можуть бути або зовнішніми, здійснюваними акторами, або внутрішніми, пов'язаними з поведінкою інших об'єктів, або тимчасовими, пов'язаними із закінченням заданого інтервалу часу.

Таблиця 21

Об'єкти, використовувані в діаграмі станів

Найменування об'єкта	Опис	Графічне позначення
Вхідна точка	Визначає подію, яка утворює початковий стан об'єкта. У точку входу не можна перейти зі стану об'єкта	
Стан	Передає ситуацію, протягом якої виконується безперервна діяльність або об'єкт перебуває в стаціонарному положенні. Стан визначається як набір значень атрибутів і відносин, пов'язаних із об'єктом. Ім'я стану повинне бути унікальним лише всередині класу об'єкта, для якого воно визначається	
Перехід станів	Визначає зміну в стані об'єкта, яка відбувається внаслідок події, що виникла в той час, коли об'єкт перебував у даному стані. Кожен перехід станів повинен мати унікальне ім'я	
Вихідна точка	Визначає завершення існування об'єкта. З точки виходу немає переходу стану	

Умова переходу – це логічний вираз, пов'язаний з атрибутами об'єкта, які повинні бути перевірені для вибору переходу стану. Умова переходу задається в тому випадку, якщо відбувається подія, внаслідок

якої може відбутися неоднозначний перехід станів. Умови переходів для одного початкового стану повинні бути взаємовиключними.

Дія – атрибут, що інформаційно описує сутність дії, яка повинна виконуватися під час переходу станів. Цій дії відповідатиме певна процедура, що реалізовує метод класу об'єктів.

Перехід станів графічно позначається лінією, на якій задається принаймні один із наступних атрибутів: виклик, умова переходу, дія.

Діаграма взаємодії об'єктів

Для кожного варіанта використання може бути побудована модель динамічної взаємодії об'єктів, яка подається однією з двох форм:

формою діаграми послідовностей (sequence diagram), що показує послідовність взаємодій на графі;

формою кооперативної діаграми (collaboration diagram), що показує взаємодію об'єктів у табличній формі.

У діаграмі послідовностей взаємодія об'єктів відтворюється у вигляді стрілки між об'єктами, яка відповідає події або повідомленню від одного об'єкта до другого, який викликає виконання об'єкта, що реагує на подію (повідомлення). Номер стрілки відповідає номеру події в послідовності.

Діаграма кооперативної поведінки подається в табличному вигляді за наступними правилами:

1. У стовпчиках таблиці вказуються об'єкти всіх типів використання, що беруть участь у реалізації варіанта. Порядок розташування активних і пасивних об'єктів довільний і повинен бути зручним для розуміння моделі. Актори прецеденту використання відображаються на правій і лівій межах таблиці.

2. По горизонталі проводяться поіменовані стрілки, що відбивають взаємодію (комунікацію) об'єктів у рамках однієї операції. Ця стрілка означає, що перший об'єкт у рамках виконаної операції посилає повідомлення другому об'єкту про необхідність виконання дії. У разі отримання повідомлення другий об'єкт виконує дію.

3. На перетині рядків і стовпчика вертикально відображається умовний відрізок часу, протягом якого виконується та або інша дія над об'єктом.

Діаграма діяльності

Діаграми взаємодій не відбивають детально порядку виконання операцій в частині розгалужень, циклічних повторень, паралельності дій.

Діаграма діяльностей передбачає відображення даних можливостей. Під **діяльністю** мається на увазі певна робота, яка може бути розбита на сукупність дій.

Діаграма діяльностей може відбивати взаємодію об'єктів з декількох варіантів використання, що, зокрема, реалізують окремо стандартні й альтернативні шляхи обробки об'єктів.

Блок, відповідний одній діяльності, може відбивати декілька подій і бути декомпозований аналогічно до блоку структурного підходу.






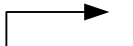

Поняття і їх графічні позначення, використовувані в діаграмі діяльностей, наведені в табл. 22.

Діаграми пакетів

В об'єктному підході пакет містить безліч взаємопов'язаних класів об'єктів і відповідає поняттю "підсистема" в структурному підході. Один варіант використання може потребувати класи об'єктів з різних пакетів. Клас об'єктів зазвичай призначається одному пакету, але з позиції досягнення різних підцілей може входити до складу різних пакетів.

Таблиця 22

Об'єкти, використовувані в діаграмі діяльностей

Найменування об'єкта	Опис	Графічне позначення
Діяльність	Робота, яка може бути розбита на сукупність дій	
Потік	Від діяльності до діяльності	
Поділ потоку	Поділ потоку на діяльності, виконувани паралельно або довільно	
Розв'язання	Вибір одного з потоків залежно від події	
Синхронізація	Перехід до однієї діяльності після виконуваних паралельно	
Ітерація	Повторне виконання діяльності	
Вихід	Визначає завершення існування	

Пакетна технологія групування класів об'єктів дозволяє спростити: розробку й експлуатацію ІС; гнучку адаптацію типових компонентів з позиції їх повторного використання; оптимізацію клієнт-серверної архітектури ІС.

Зазвичай ІС розбивається на функціональні й забезпечувальні пакети. Функціональні пакети, відповідні вирішуваним проблемам (завданням), об'єднуються в один спільний пакет "Предметна область". Кожен пакет, у свою чергу, може бути розбитий на підпакети відповідно до семантичної близькості й щільності взаємодії класів об'єктів. Зазвичай пакети предметної області містять ієрархії узагальнення й агрегації. Класи об'єктів, потрібні в декількох підсистемах, виділяються в самостійні пакети. В одному пакеті визначається не більше 20 компонентів, зазвичай 5 – 15.

Із точки зору забезпечувальних пакетів ІС розбивають на п'ять основних пакетів:

інтерфейс, об'єкти якого реалізують функції взаємодії користувачів з ІС щодо введення-виведення інформації й обміну повідомленнями між підсистемами;

база даних, об'єкти якої виконують доступ до даних у зовнішній пам'яті;

управління завданнями, об'єкти якого здійснюють функції диспетчеризації і маршрутизації обробки об'єктів, наприклад у системі управління робочими потоками;

утиліти, об'єкти якого здійснюють допоміжні функції, наприклад перетворення форматів даних;

забезпечувальні пакети, тобто ті, що працюють за принципом "клієнт-серверної" архітектури, виконують серверні функції для функціональних об'єктів-клієнтів. Таким чином, забезпечувальні пакети звільняють користувача від знання деталей програмно-технічної реалізації ІС.

Діаграми компонентів і розміщення

Діаграма компонентів відображає залежності програмних компонентів, які подаються у вигляді початкових програмних кодів об'єктів, що відкомпілювалися, і виконуваних кодів. Один компонент, як правило, відповідає програмному коду одного пакета класів об'єктів.

Компонент у своєму складі має інтерфейсний клас об'єктів, через який здійснюється доступ до решти класів об'єктів компоненти. За допомогою інтерфейсу об'єкти інших компонентів звертаються не до конкретних об'єктів даного компонента, а до його інтерфейсного об'єкта. Таким чином спрощується взаємодія компонентів між собою, коли в разі доступу до компоненти з інших компонентів не потрібно знати внутрішню структуру цієї компоненти. Компонент, до якого здійснюється звернення, може бути не об'єктно-орієнтованою. Достатньо, щоб у такого компонента був лише один інтерфейсний клас

об'єктів, який транслює запити до компонента у виклики звичайних процедур. У компонентів може бути декілька інтерфейсів.

11.3.3. Технологія об'єктно-орієнтованого проектування

Слід розглянути детальніше зміст операцій, що виконуються у випадку об'єктного підходу до проектування.

1. Аналіз системних вимог.

Аналіз системних вимог здійснюється на основі опису предметної області, отриманої в ході робіт з аналізу й проектування бізнес-процесів. Ці матеріали містять опис організаційної структури, структури матеріальних, фінансових і інформаційних потоків, який може бути виконаний як традиційними графічними методами, так і за допомогою об'єктно-орієнтованої методології. На рис. 43 наведена технологія аналізу системних вимог.

2. Логічне проектування.

На етапі логічного проектування здійснюється деталізація моделей варіантів використання, класів об'єктів, станів, пакетів і розробка моделей взаємодії об'єктів і діяльностей, які визначають характер методів (процедур) обробки об'єктів. Технологія логічного проектування ІС наведена на рис. 44.

3. Фізичне проектування.

На етапі фізичного проектування відбувається деталізація діаграм класів об'єктів і пакетів для їх подальшої реалізації в конкретному програмно-технічному середовищі. Технологія фізичного проектування наведена на рис. 45.

На етапі реалізації ІС здійснюється генерація кодів класів об'єктів, програмування процедур методів класів об'єктів, заповнення баз даних розміщення компонентів у вузлах обчислювальної мережі (рис. 46).

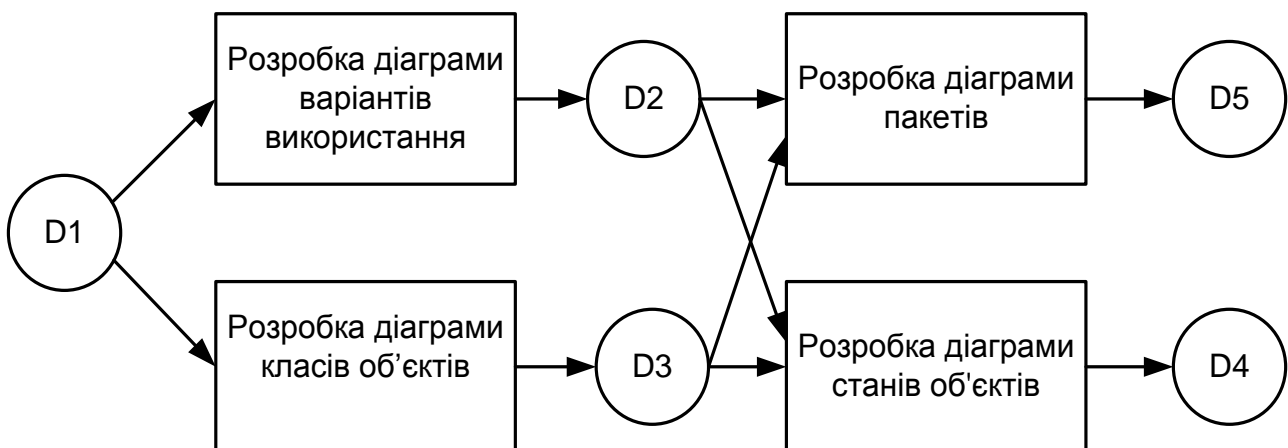


Рис. 43. Технологія аналізу системних вимог

Умовні позначення:

- D1 – опис предметної сфери;
- D2 – діаграма варіантів використання;
- D3 – діаграма класів об'єктів;
- D4 – діаграма станів об'єктів;
- D5 – діаграма пакетів.

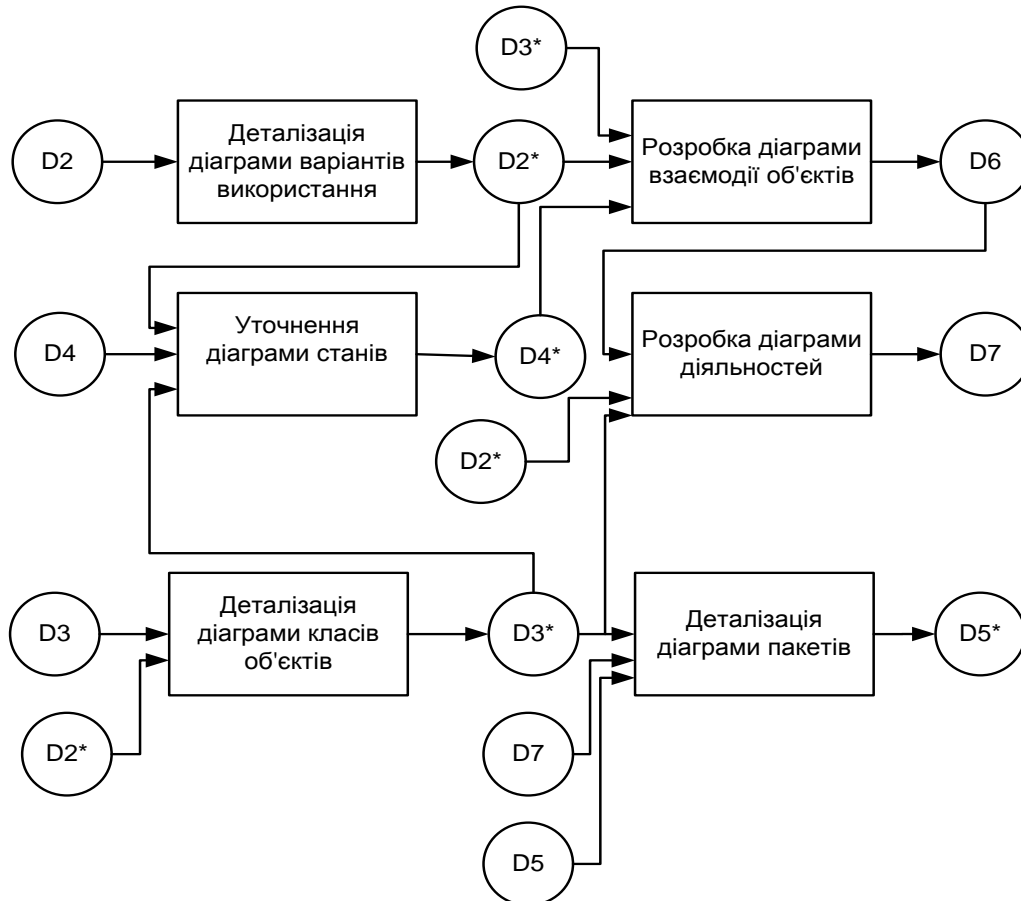


Рис. 44. Технологія логічного проектування

Умовні позначення:

- D2, D2* – діаграма варіантів використання;
- D3, D3* – діаграма класів об'єктів;
- D4, D4* – діаграма станів об'єктів;
- D5, D5* – діаграма пакетів;
- D6 – діаграма взаємодії;
- D7 – діаграма діяльностей.

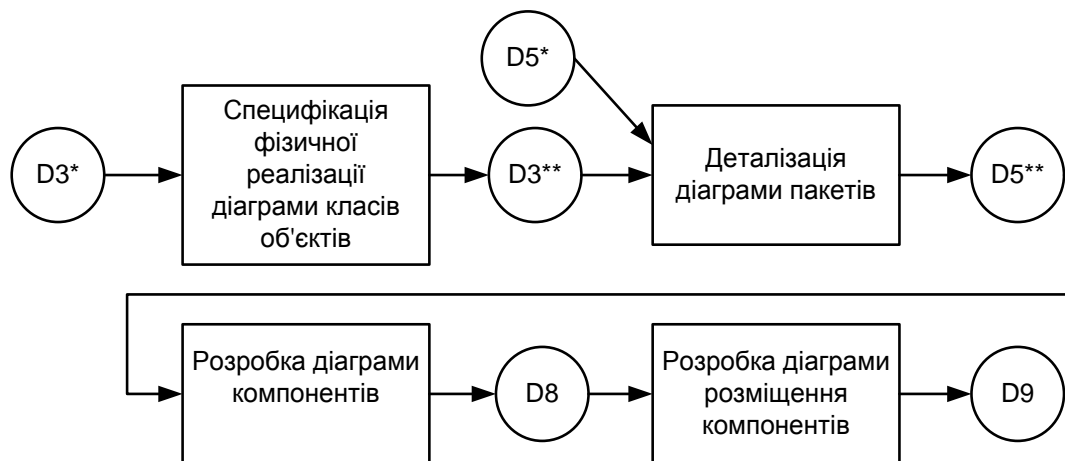


Рис. 45. Технологія фізичного проектування

Умовні позначення:

D3*, D3** – діаграма класів об'єктів;

D5*, D5** – діаграма пакетів;

D8 – діаграма компонентів;

D9 – діаграма розміщення компонентів.

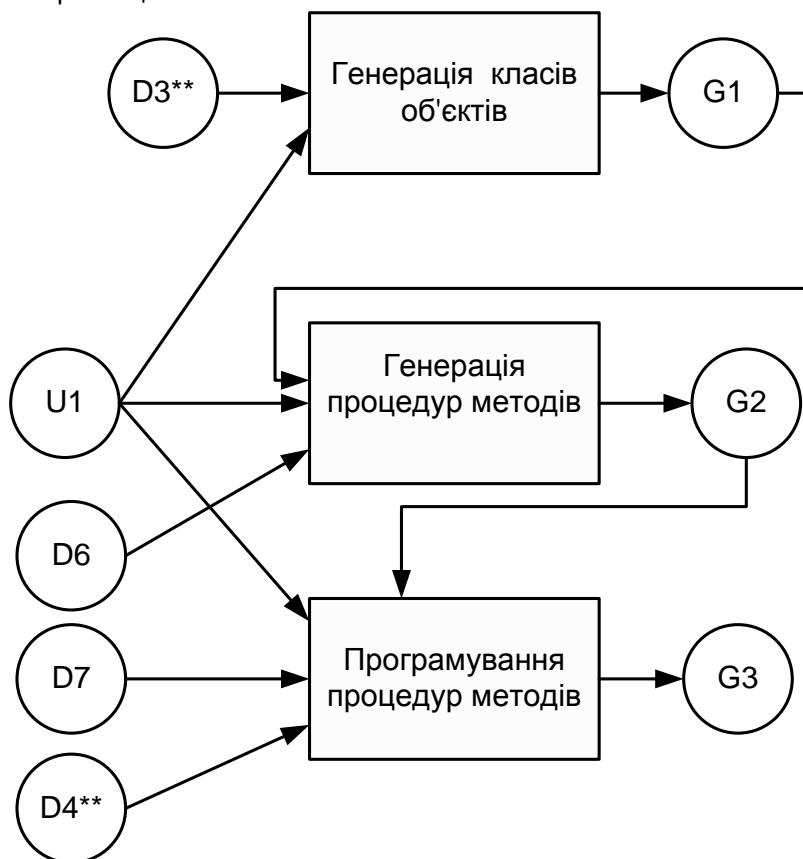


Рис. 46. Технологія реалізації ІС

Умовні позначення:

D3* – діаграма класів об'єктів;

D4* – діаграма станів об'єктів;

D6 – діаграма взаємодії;

D7 – діаграма діяльностей;

U1 – універсум об'єктно-орієнтованих мов програмування;

- G1 – класи об'єктів;
- G2 – шаблони процедур методів класу об'єктів;
- G3 – процедури методів.

Контрольні запитання

1. Охарактеризуйте архітектуру CASE-систем.
2. Розкрийте поняття предметної області, моделі предметної області.
3. Чому потрібно проводити попереднє моделювання предметної області?
4. Які вимоги висуваються до моделей предметної області?
5. Які моделі відбивають структурні аспекти предметної області?
6. Охарактеризуйте оцінні аспекти моделювання предметної області.
7. На яких рівнях будуються моделі предметної області?
8. Охарактеризуйте об'єктну структуру на трьох рівнях деталізації.
Що таке об'єкт, динамічний об'єкт, статичний об'єкт? Наведіть приклади.
9. Охарактеризуйте функціональну структуру на трьох рівнях деталізації. Що таке функція? Наведіть приклади.
10. Охарактеризуйте структуру управління на трьох рівнях деталізації. Що таке подія? Наведіть приклади.
11. Що таке організаційна структура, організаційна одиниця? Які існують різновиди організаційних структур? Чим вони відрізняються?
12. Охарактеризуйте організаційну структуру управління на трьох рівнях деталізації. Наведіть приклади.
13. Охарактеризуйте технічну структуру на трьох рівнях деталізації.
Що таке топологія, комунікація?
14. У чому полягає сутність структурного підходу до проектування?
15. Що є базовим елементом структурного підходу?
Охарактеризуйте його.
16. Яка основна мета структурного аналізу?
17. Які характерні риси і принципи структурного аналізу?
18. Які нотації та інструментальні засоби використовуються в методологіях структурного підходу?
19. Наведіть класифікацію структурних методологій.
20. Охарактеризуйте методологію функціонального моделювання у структурному підході.
21. Охарактеризуйте методології моделювання потоків даних у структурному підході.

22. Охарактеризуйте методологію моделювання даних у структурному підході.
23. Охарактеризуйте методологію моделювання потоків робіт у структурному підході.
24. Які роботи виконуються під час структурного проектування?
25. Охарактеризуйте структурні карти, що використовуються в структурному проектуванні.
26. У чому полягає сутність об'єктного підходу до проектування?
27. Які поняття використовують в об'єктному підході? Дайте їх визначення.
28. Охарактеризуйте принципи об'єктного підходу.
29. Які типи моделей використовуються в об'єктному підході.
30. Що є стандартом об'єктного проектування? Хто його розробив? У яких CASE-технологіях цей стандарт використовується?
31. Які моделі включає нотація UML? Охарактеризуйте їх.
32. Які особливості технології проектування за об'єктного підходу?
33. Розкрийте зміст операцій, що виконуються за об'єктного підходу для аналізу системних вимог. Які моделі використовуються на цій фазі?
34. Розкрийте зміст операцій, що виконуються за об'єктного підходу для логічного проектування. Які моделі використовуються на цій фазі?
35. Розкрийте зміст операцій, що виконуються за об'єктного підходу для фізичного проектування. Які моделі використовуються на цій фазі?
36. Розкрийте зміст операцій, що виконуються за об'єктного підходу для реалізації ІС.

12. Комбінований процесно-орієнтований підхід до проектування ІС

12.1. Передумови виникнення підходу

Висока вартість сучасних інформаційних систем і їх вплив на ефективність діяльності підприємств вимушують вкладати чималі кошти в технології, що дають упевненість в кінцевому результаті. Тому останнім часом під час проектування великих ІС використовується так званий комбінований процесно-орієнтований підхід. Передумовами його розвитку і широкого використання стало наступне.

1. Орієнтація на впровадження процесного підходу до управління організаціями.

Серед менеджерів усіх рівнів сформувалася думка, що основні резерви в підвищенні ефективності бізнесу містяться саме в області оптимізації бізнес-процесів. Вислів засновників процесного підходу до управління організаціями М. Чампі і Дж. Хаммера визначив сучасний погляд на управління бізнесом: "Не товари, а процеси їх створення приносять компаніям довгостроковий успіх". Зусилля менеджменту фокусуються на управлінні не окремими функціями підприємства, а бізнес-процесами, що визначають суть його діяльності. Процесний підхід в управлінні дозволяє підвищити конкурентоспроможність компанії, бути гнучкішим і адекватнішим до змін на ринку, принципово поліпшити якість продуктів і послуг. Цей підхід спрямований на ліквідацію фрагментарності в роботі організації, інформаційних і організаційних розривів, дублювання функцій, нераціонального використання матеріальних і людських ресурсів, скорочення операційних витрат.

2. Наявність у традиційних підходів до проектування (структурних і об'єктних) як переваг, так і недоліків привела до необхідності поєднання моделей різних підходів.

Кожна з моделей має унікальні властивості, відсутні в інших. Тому для поліпшення якості моделювання в комбінованому підході використовуються моделі різних підходів. Зручне поєднання моделей різних підходів покращує якість системних специфікацій. Уся сукупність моделей розглядається як єдиний набір інструментів. У процесі побудови кожної інформаційної системи потрібно вибирати ті інструменти, які найбільше підходять в даному конкретному випадку, не обмежуючи себе рамками одного методу, яким би популярним він не був.

3. Використання сучасних інформаційних технологій в організаційному управлінні як чинник розвитку методологій та інструментальних засобів реінжинірингу бізнес-процесів (РБП).

РБП – це інженерний підхід до проектування й управління організацією. Він передбачає спільну участь професіоналів предметної сфери і розробників ІС у створенні ІС. Для знаходження між ними спільної мови відбувається інтеграція сучасних інформаційних технологій: інженерія знань, імітаційне моделювання, CASE-технології та інші.

12.2. Основні поняття процесного підходу

Поняття "процес" – ключове в сучасній теорії управління бізнесом. Відповідно до міжнародного стандарту ISO 9000:2000 **процес** – це сукупність взаємопов'язаних і взаємодіючих видів діяльності, що перетворюють входи у виходи. При цьому слово "діяльність", вживане у визначенні, має своє звичайне значення, наведене в словниках.

Назви процесів, підпроцесів повинні бути виражені дієсловом або віддієслівним іменником, наприклад, процес постачання, процес виробництва, процес маркетингу і т. д. Процес включає одну або більше пов'язаних між собою процедур (дій), які спільно реалізують певне завдання бізнесу, – зазвичай у рамках організаційної структури. Він може виконуватися в межах однієї організаційної одиниці, охоплювати декілька одиниць або навіть декілька різних організацій, наприклад, у системі "покупець – постачальник". Процес зазвичай пов'язаний з операційними відносинами, наприклад, процес розробки нового виробу або процес продажів.

Досягнення певної сукупності цілей за рахунок виконання бізнес-процесів називається **деревом цілей**. Дерево цілей має, як правило, ієрархічний вигляд. Кожна мета має свою вагу і критерій (кількісний або якісний) досяжності.

Бізнес-процеси реалізують бізнес-функції підприємства. Під бізнес-функцією розуміють вид діяльності підприємства. Множина бізнес-функцій становить ієрархічну декомпозицію функціональної діяльності і називається **деревом функцій**.

Бізнес-функції пов'язані з показниками діяльності підприємства, що створюють **дерево показників**. На основі дерева показників будується система показників оцінки ефективності виконання процесів. Власники процесів контролюють свої бізнес-процеси за допомогою даної системи показників. Найбільш загальними показниками оцінки ефективності бізнес-процесів є:

- кількість продукції заданої якості, яка виготовляється за певний інтервал часу;

- кількість споживаної продукції;

- тривалість виконання типових операцій та ін.

- Слід ввести певні терміни, пов'язані з поняттям процесу.

Кожен бізнес-процес має свої ролі й межі.

У процесному підході використовуються наступні ключові ролі: власник процесу, лідер команди, комунікатор, координатор процесу, учасник команди.

У стандарті ISO 9000:2000 зазначено, що слід призначити особу, яка відповідає за процес, тобто її власника. **Власник процесу** – особа (або група осіб), що відповідає за процес і має повноваження змінювати його з метою вдосконалення. Власник процесу має у своєму розпорядженні ресурси, необхідні для виконання процесу і відповідає за результат процесу. Власник управляє процесом і є невід'ємною складовою його частиною. Він повинен знати бізнес-процес, стежити за його виконанням і вдосконалювати його ефективність. Власникові бізнес-процесу необхідно володіти комунікативністю, ентузіазмом, здатністю впливати на людей і проводити зміни. Схема взаємодії процесу і його власника наведена на рис. 47. На схемі видно, що власник процесу є частиною самого процесу.

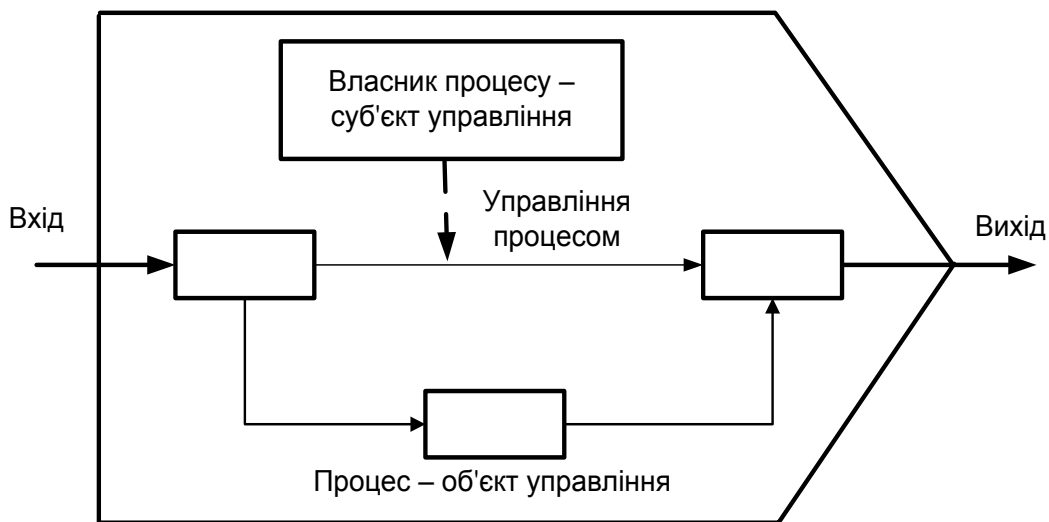


Рис. 47. Взаємодія процесу і його власника

Лідер команди – працівник, що володіє знаннями про бізнес-процес і має позитивні особисті якості.

Комунікатор – працівник, що навчає команду різним методам роботи, готує спільно з лідером наради і аналізує їх результат.

Координатор процесу – працівник, що відповідає за погоджену роботу всіх частин бізнесу і забезпечує зв'язок з іншими бізнес-процесами. Координатор повинен мати адміністративні здібності й розуміння стратегічних цілей підприємства.

Учасники команди – фахівці різних рівнів ієрархії. Учасники команди отримують підтримку і методичне забезпечення від

консультанта і комунікатора, разом з лідером проводять моделювання, аналіз і оцінку бізнес-процесу.

Одним з основних елементів процесного підходу є команда. Існує декілька типів процесних команд.

Ситуаційна команда зазвичай працює на постійній основі і виконує роботу, що періодично повторюється.

Віртуальна команда створюється для розробки нового продукту або послуги.

Ситуаційний менеджер – висококваліфікований фахівець, здатний самостійно виконати до 90% обсягу робіт.

Межі процесу пов'язані з такими поняттями, як вихід, вхід, ресурс, інтерфейс.

Вихід (продукт) процесу – матеріальний або інформаційний об'єкт або послуга, що є результатом виконання процесу і споживаний зовнішніми відносно процесу клієнтами. Вихід процесу завжди має споживача. Якщо споживачем є інший процес, то для нього цей вихід є входом. Виходом може бути продукція, документація, інформація, персонал, послуги і т. д.

Вхід процесу – продукт, який у ході виконання процесу перетворюється у вихід. Вхід завжди повинен мати постачальника. Входом може бути: сировина, матеріали, напівфабрикати, документація, інформація і т. д. Входом до процесу зазвичай є виходи інших процесів.

Ресурс процесу – матеріальний або інформаційний об'єкт, постійно використовуваний для виконання процесу, але який не є входом процесу. Ресурсом може бути інформація, персонал, устаткування, програмне забезпечення, інфраструктура, середовище, зв'язок, транспорт і т. д.

Межі процесу – це межа входу і межа виходу. Межа входу передуює першій операції процесу, межа виходу йде за його останньою операцією.

Інтерфейс процесу – механізм (організаційний, інформаційний, технічний), за допомогою якого процес взаємодіє з попереднім і наступним процесами.

Виходячи з наведених вище визначень, схему управління процесом можна подати таким чином (рис. 48).

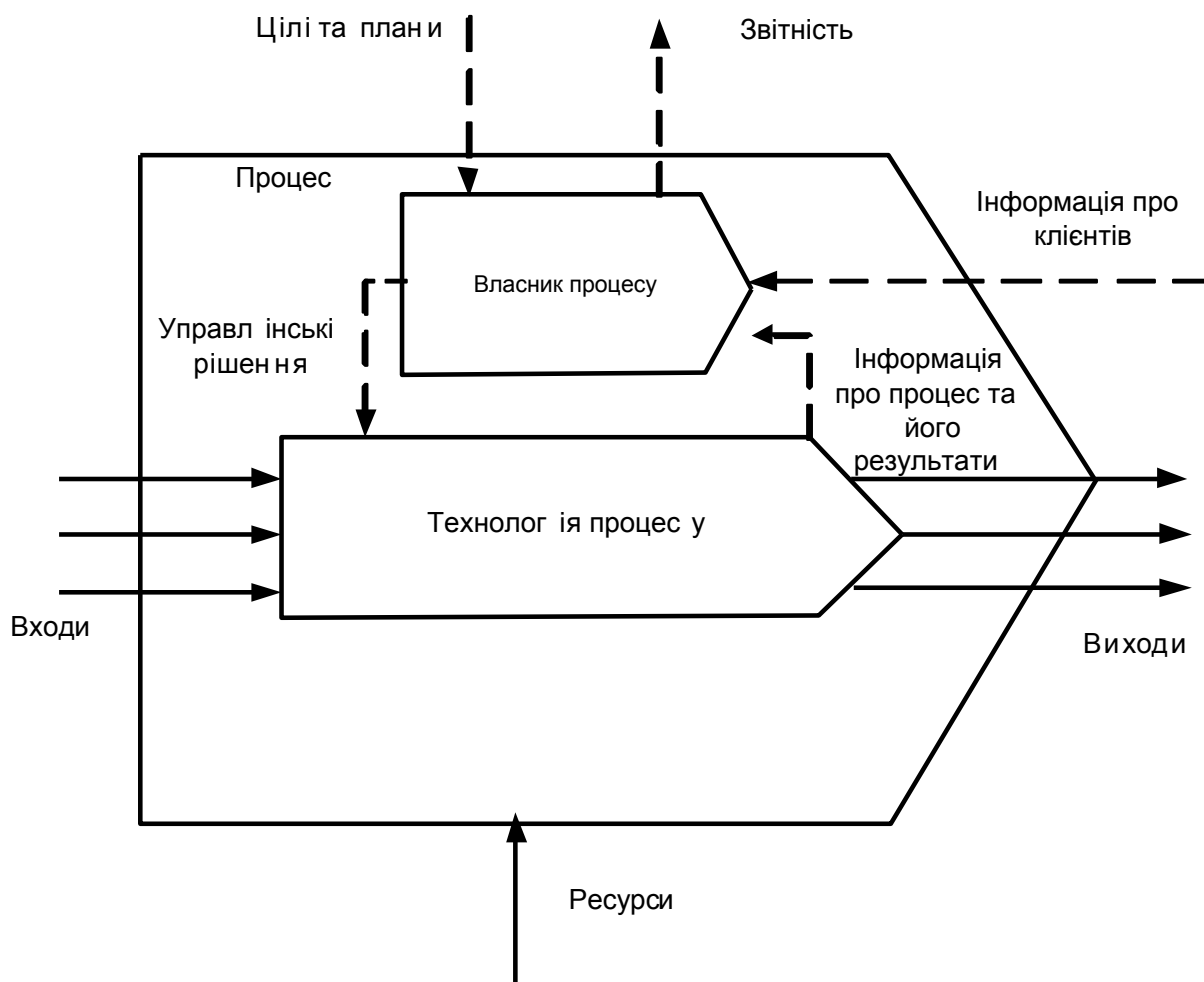


Рис. 48. Схеми управління процесом

Умовні позначення:

- > Потоки продуктів і ресурсів;
- - - - -> Потоки інформації, управлінські рішення

Відмінності між ресурсами і входами процесу наведені в табл. 23. Виходи, входи, ресурси позначаються іменниками, оскільки вони є матеріальними об'єктами.

Процес має входи і виходи. Для його виконання використовуються ресурси: персонал, устаткування, інфраструктура і т. д. Управління процесом здійснює його власник. Виконання процесу здійснюється за допомогою ресурсів, що перебувають у розпорядженні власника процесу.

Таблиця 23

Порівняльна характеристика ресурсів і входів процесу

Ресурси	Входи
Управляються власниками процесу	Надходять в процес із зовні
Обсяг ресурсів планується на багато	Обсяг входів планується на один або

циклів роботи або тривалий період роботи процесу	декілька циклів роботи процесу
--	--------------------------------

Процес не може існувати окремо від організації. Для виконання процесів вище керівництво повинне визначити призначення процесу, поставити перед власником процесу мети, затвердити планові показники результативності і ефективності процесу. Власник процесу ухвалює управлінські рішення на підставі інформації, що надійшла, і встановлених планів.

12.3. Основи реінжинірингу бізнес-процесів

12.3.1. Реінжиніринг бізнесу як основа сучасного менеджменту організації

Сучасний підхід до управління підприємством ґрунтується на таких концепціях.

1. Інтеграція управлінських та інформаційних технологій. Інформаційні технології стають ефективним інструментом вдосконалення процесів управління, невід'ємною складовою самих процесів.

2. Автоматизоване управління будується на принципах, що відрізняються від традиційного управління організаціями, і вимагає корінної перебудови всієї системи управління. Процес впровадження інформаційної системи в організації тісно пов'язаний із перебудовою самої системи управління – оптимізацією організаційної структури, функцій і процесів, які описують взаємодію ланок цієї структури, а також зі зміною мотивації персоналу.

До початку 80-х років в управлінні організаціями використовувався так званий функціональний підхід. Його суть полягає в управлінні функціями. Уся діяльність організації розбивається на однорідні та спеціалізовані функції, які зосереджувалися в окремих підрозділах організації. Підрозділи об'єднуються в ієрархічні структури. Управління такою структурою фактично становить управління функціями. Цей підхід до управління має ряд переваг:

- 1) зрозумілість для співробітників структурних підрозділів;
- 2) простота управління по горизонталі та вертикалі.

Функціональний підхід виправдовував себе в умовах таких чинників розвитку організацій, як гарантовані ринки збуту, попит на продукти, що постійно збільшується, і послуги. Недолік функціонального підходу полягає в тому, що в ньому процеси діляться на функції, які об'єднуються потім в

рамках структурного підрозділу. Подальше управління спирається лише на досвід, на емпірично виділені функції, а не на ланцюги дій (процесів), що спрямовані на кінцевий результат. Основними недоліками функціонального підходу є наступні:

- розбиття технологій виконання роботи на окремі фрагменти, іноді між собою не пов'язані, які виконуються різними структурними підрозділами;
- відсутність цілісного опису технологій виконання роботи;
- складність пов'язування простих завдань в технологію, що виробляє реальний товар або послугу;
- відсутність відповідальності за кінцевий результат;
- високі витрати на узгодження, налагодження взаємодії, контроль і т. д.;
- відсутність орієнтації на клієнта.

Зростання конкурентної боротьби спричиняло необхідність пошуку організаціями шляхів радикального підвищення ефективності своєї діяльності. Це привело до появи нових підходів до управління, серед яких найбільшого поширення набув процесний підхід.

Суть процесного підходу полягає в наступному. Уся діяльність організації складається з двох видів дій:

- повторюваних, які слід здійснювати періодично або в разі настання певних подій. Цей вид дій називається процесами;
- разових, унікальних за складом, таких, що не повторюються в такому вигляді надалі. Цей вид діяльності називається проектами, програмами, заходами і т. д.

Виходячи з цього, управління діяльністю компанії включає:

- управління процесами (Process Management);
- управління проектами (Project Management).

Основну частку діяльності організації складають саме процеси. Процесний підхід означає управління власне діяльністю на відміну від управління її структурою, як у функціональному підході.

Сучасний стан економіки характеризується саме переходом від традиційної функціональної моделі діяльності організації, побудованої на принципах розподілу праці, вузькій спеціалізації і жорстких ієрархічних структурах, до процесної моделі, заснованої на інтеграції робіт навколо бізнес-процесів.

Відповідно до нового погляду на організацію робота повинна бути організована навколо процесів. **Метою** організації повинне бути вдосконалення бізнес-процесів для подолання їх фрагментарності і для

досягнення суттєвого поліпшення в ключових показниках результативності – витратах, якості, рівні обслуговування і оперативності.

Процесний підхід передбачає зміщення акцентів з управління окремими структурними елементами на управління кризовими бізнес-процесами, що пов'язують діяльність усіх структурних елементів. Кожен діловий процес проходить через ряд підрозділів, тобто в його виконанні беруть участь фахівці різних відділів компанії. Найчастіше доводиться стикатися з ситуацією, коли власне процесами ніхто не управляє, а управляють лише підрозділами. Більше того, структура компаній будується без урахування можливостей оптимізації ділових процесів, що забезпечують виконання необхідних функцій. Процесний підхід дозволяє усунути фрагментарність у роботі, організаційні й інформаційні розриви, дублювання, нераціональне використання фінансових, матеріальних і кадрових ресурсів.

Процесний підхід до організації діяльності підприємства передбачає:

- широке делегування повноважень і відповідальності виконавцям;
- скорочення рівнів ухвалення рішень;
- поєднання принципу цільового управління з груповою організацією праці;
- підвищену увагу до питань забезпечення якості;
- автоматизацію технологій виконання бізнес-процесів.

Бізнес-процеси (БП) є сукупністю взаємопов'язаних дій (операцій, процедур, робіт) з виготовлення готової продукції або виконання послуг на основі споживання ресурсів. Бізнес-процеси – це ділові, адміністративні, технологічні процедури функціонування підприємства, до яких належать: документообіг, управління фінансовими, матеріальними потоками, персоналом, організаційно-господарськими й технологічними процесами, процесами проектування виробів тощо. Управління бізнес-процесами спрямоване на виконання якісного обслуговування споживачів (клієнтів). При цьому в ході управління бізнес-процесами всі матеріальні, фінансові й інформаційні потоки розглядаються у взаємодії.

Незважаючи на очевидні переваги процесно-орієнтованої побудови компанії, досягти створення такої структури в чистому вигляді досить важко.

Процес зміни системи управління підприємством є багатоетапним.

На **першому етапі** визначають місію (основне призначення) організації, її стратегічні цілі. Поняття **місія** згідно зі стандартом ІСО-15704 – це:

діяльність, здійснювана підприємством для того, щоб виконати функцію, для якої воно було засноване, – надання замовникам продукту або послуги;

механізм, за допомогою якого підприємство реалізує свої цілі та завдання.

Завдання першого етапу вирішуються, виходячи з аналізу зовнішнього середовища організації. Даний етап виконують фірми, які займаються управлінським консалтингом і аудитом.

Другий етап – аналіз і адаптація внутрішнього середовища організації з тією метою, щоб її структура і принципи функціонування відповідали місії підприємства і були спрямовані на досягнення поставлених стратегічних цілей. Цей етап називається **реінжинірингом бізнес-процесів**. Якщо ж необхідне впровадження автоматизованої системи підтримки бізнесу, то даний етап можна назвати визначенням вимог до інформаційної системи (аналізом вимог), коли на основі цільових моделей діяльності підприємства (моделей "як повинно бути") виявляються об'єктивні вимоги до тих завдань, виконання яких повинна забезпечувати автоматизована система, що розробляється.

Метою реінжинірингу бізнес-процесів є системна реорганізація матеріальних, фінансових та інформаційних потоків, спрямована на спрощення організаційної структури, перерозподіл і мінімізацію використання різних ресурсів, скорочення термінів реалізації потреб клієнтів, підвищення якості їх обслуговування.

Першим кроком проекту щодо оптимізації діяльності повинні стати виділення основних продуктів компанії і вибудовування процесів відповідно до продуктових ліній. Це дозволяє отримати продуктові "зрізи" бізнес-процесів, що відбуваються в організації. Однак завжди існує декілька функціональних підрозділів, які беруть участь в обслуговуванні всіх продуктових ліній, наприклад, бухгалтерія, транспортний цех і т. д. Перебудувати дані напрями, розбивши їх на продуктові зрізи компанії, надто складно, оскільки це спричинить масу додаткових проблем і витрат.

Таким чином, завдання формалізації і оптимізації діяльності зводиться до виділення бізнес-процесів відповідно до продуктових ліній і функціональних підрозділів, і узгодження їх в наскрізні процеси компанії, націлені насамперед на створення продуктів і надання послуг клієнтам.

Інжиніринг бізнес-процесів включає реінжиніринг бізнес-процесів, що проводиться з певною періодичністю, наприклад один раз на 5 – 7 років, і подальше безперервне поліпшення бізнес-процесів шляхом їх адаптації до зовнішнього середовища, що змінюється. Інжиніринг бізнес-процесів виконується на основі застосування інженерних методів і сучасних програмних

інструментальних засобів моделювання бізнес-процесів спільними командами фахівців компанії і консалтингової фірми.

12.3.2. Фази реінжинірингу бізнес-процесів

Реінжиніринг бізнес-процесів полягає в узгодженій оптимізації головних складових системи управління: організаційної структури, функцій, процесів і ресурсів. Щоб оптимізувати систему управління, потрібно її багато разів переконфігурувати, порівняти конструкції, які утворилися, між собою і вибрати найкращу. Виконати це без інструментальної підтримки з боку комп'ютерних технологій неможливо.

У моделях життєвого циклу ІС, передбачалося, що вимоги до організації ІС, як правило, чітко визначаються до моменту початку проектування ІС, тобто організаційна система повинна бути спроектована, а ще краще – повинна функціонувати до моменту розробки ІС.

Реінжиніринг бізнес-процесів передбачає, що реорганізація організації не може бути успішно проведена без створення адекватної ІС. Тому ІС не просто автоматизує наявні ділові процеси "як є", а забезпечує підтримку змін організації на принципах "як повинно бути". Унаслідок цього реорганізація організації і проектування ІС відбуваються практично паралельно.

Паралельність життєвого циклу розробки ІС означає також те, що більшість основних бізнес-процесів, що реорганізуються, проектуються одночасно, що викликає необхідність паралельної координації проведених робіт у частині розробки загальних забезпечувальних підсистем. Таким чином, загальносистемні рішення формуються в процесі реалізації вимог за окремими бізнес-процесами. Послідовність етапів проведення бізнес-реінжинірингу наведена на рис. 49.

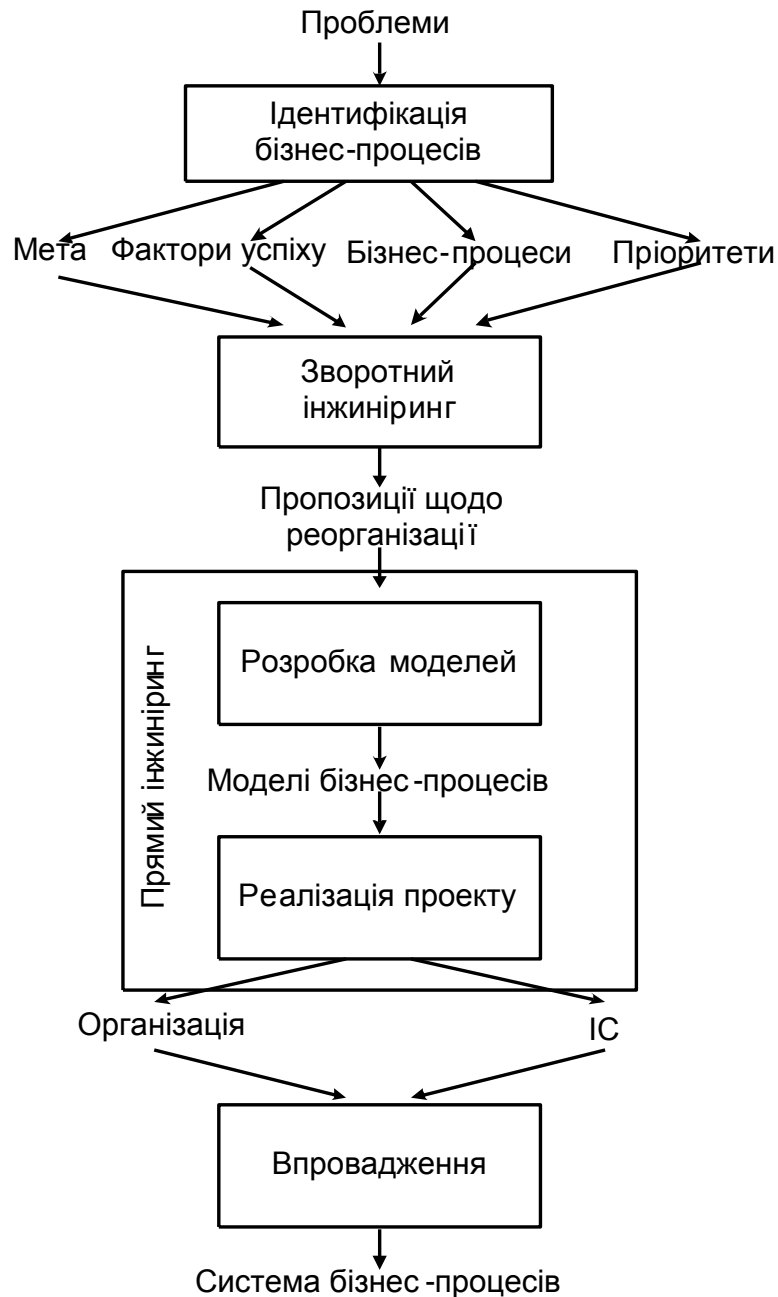


Рис. 49. Фази проведення бізнес-реінжинірингу

Слід розглянути фази проведення реінжинірингу бізнес-процесів більш докладно.

1. Ідентифікація бізнес-процесів.

На фазі ідентифікації виділяються ключові бізнес-процеси, реорганізація яких забезпечує кардинальне підвищення ефективності функціонування організації.

Роботи з ідентифікації бізнес-процесів ініціюють менеджери верхньої ланки управління підприємством – особи, що ухвалюють рішення. На цьому етапі здійснюється постановка завдань реінжинірингу бізнес-процесів, яка

визначається ключовими проблемами підприємства, що викликають необхідність РБП. Наприклад, у зв'язку зі зниженням обсягу продажів або збільшенням кількості реклаमाцій на продукцію, або високою плинністю кадрів, або низькою завантаженістю устаткування, або міжопераційними простоями, або наднормативними запасами і т. д.

У процесі постановки реінжинірингу бізнес-процесів вирішуються наступні завдання:

формулювання (уточнення) місії підприємства;

визначення критичних факторів успіху (7 – 8 факторів) чи критеріїв ефективності організації бізнес-процесів: тривалості, витрат, якості, сервісного обслуговування і т. д.;

виявлення основних видів бізнес-процесів, які вже існують, так і перспективних (10 – 15 процесів);

неформальний опис відмінних рис виявлених бізнес-процесів;

оцінка важливості бізнес-процесів за ступенем впливу на реалізацію критичних факторів успіху;

визначення обмежень, пов'язаних із рівнем кваліфікації персоналу фірми, технічної оснащеності виробництва, наявністю інформаційних технологій, фінансових ресурсів;

опис можливих сценаріїв розвитку підприємства: поява нових технологій, ресурсів, зміна поведінки клієнтів, партнерів, конкурентів;

визначення зовнішніх ризиків забезпечення фінансовими ресурсами, надійності партнерів, кон'юнктури ринку;

ранжування бізнес-процесів для проведення реінжинірингу бізнес-процесів відповідно до отриманої оцінки важливості, обмежень, ризиків і перспектив.

Для підготовки ухвалення рішень щодо проведення реінжинірингу процесів чи уточнення сформульованих реінжинірингом бізнес-процесів завдань може бути проведена робота з узагальненого аналізу фінансово-господарської діяльності підприємства з використанням сучасних інформаційних технологій.

Так, **формулювання місії** передбачає вирішення завдань якісного аналізу ділових процесів, пов'язаних із визначенням стратегії поведінки підприємства на ринку в частині розширення меж ринку чи глибокого проникнення на ринок, диверсифікованості діяльності чи підвищення якості товарів і послуг, глобалізації чи локалізації діяльності і т. д. Як основні методи формування стратегії підприємства звичайно

використовуються методи аналізу ієрархій Сааті, нечіткої логіки Заде, а з інструментальних засобів – статичні експертні системи з можливістю обробки якісних (нечітких) оцінок, такі як Expert Choice, Guru, Level5, Nextpert Objects та ін.

Оцінка обмежень і ризиків, пов'язаних із проведенням РБП, виконується в ході вирішення завдань бізнес-планування. Зокрема, для виділених бізнес-процесів здійснюється оцінка можливостей підприємства в плані ефективності розподілу капіталовкладень для різних проектів. Для вирішення цього завдання звичайно використовуються економіко-математичні моделі й методи оптимізації. До найбільш відомих програмних засобів бізнес-планування відносяться ППП Project Expert, "Альт-Инвест", ТЭО-ИНВЕСТ, COMFAR PROSPIN та ін.

Вибір бізнес-процесів виконується на основі аналізу сегментів ринку, за допомогою якого конкретизуються стратегічні цілі підприємства для визначення регіонів, споживачів, каналів розподілу продукції та послуг, а отже, й особливості організації бізнес-процесів. Основними методами досліджень на цьому етапі виступають методи статистичного аналізу і прогнозування, нейронних мереж, інтелектуального аналізу даних сучасних інформаційних сховищ. Найбільш могутніми інструментальними засобами аналізу і прогнозування для виявлення основних сегментів ринку є ППП SAS, Oracle Express, Business Objects, SPSS, NeurOn-Line, Brain Maker, Poly-Analyst та ін.

Після ухвалення рішення про проведення РБП виконується планування робіт на всіх стадіях життєвого циклу проекту, виділяються матеріальні, людські, фінансові й часові ресурси, формується організаційна структура проекту і команди РБП, проводиться роз'яснювальна робота з персоналом підприємства з питань майбутньої реорганізації підприємства.

Планування РБП здійснюється за допомогою методів і засобів управління проектами. З використанням програмних засобів управління проектами на зразок TimeLine, Microsoft Project, Primavera будуються мережні календарні графіки виконання робіт. Більш складні програмні продукти, такі, наприклад, як Process Engineer (LBMS), дозволяють визначати ресурси на виконання проектних робіт, вести бібліотеки проектів, здійснювати колективну розробку і моніторинг стану проекту на наступних стадіях РБП. Засоби планування та управління роботами підтримуються також у комплексних технологіях проектування і

впровадження корпоративних ІС, наприклад CASE Method (Oracle), Accelerated SAP (SAP), Orgware (BAAN).

2. Зворотний інжиніринг.

Зворотний інжиніринг передбачає дослідження бізнес-процесів, які функціонують на підприємстві. Мета етапу полягає в проведенні діагностики "вузьких місць" в організації наявних бізнес-процесів і формулюванні напрямків їх реорганізації. Завдання зворотного інжинірингу спрощується, якщо на підприємстві є документація про процеси, які функціонують після попередньої реорганізації.

На етапі зворотного інжинірингу постановка завдань реорганізації бізнес-процесів уточнюється, сформульовані на фазі ідентифікації бізнес-процесів у загальному вигляді цілі РБП можуть бути скоректовані за результатами дослідження наявної системи організації бізнес-процесів.

Зворотний інжиніринг не повинен викликати одержання надмірно детальної картини існуючих бізнес-процесів, тому що в цьому випадку велика ймовірність не вирішити поставлене завдання. На фазі зворотного інжинірингу будуються, як правило, тільки принципові моделі бізнес-процесів, що дозволяють зрозуміти сутність функціонування підприємства в цілому і виявити напрямки реорганізації бізнес-процесів. За необхідності уточнення деталей яких-небудь ділових процесів до зворотного інжинірингу можна ітеративно повернутися на фазі прямого інжинірингу.

На фазі зворотного інжинірингу для розуміння сутності бізнес-процесів, які функціонують, широко використовуються методи і засоби структурного аналізу ділових та інформаційних процесів (функціонального чи об'єктного моделювання), які реалізовані в таких відомих ППП, як, наприклад, Design/IDEF, BPWin, ERWin, OOWin, ARIS Toolset, Rational Rose та ін.

Для оцінки ефективності наявних бізнес-процесів використовуються, насамперед, методи і засоби функціонально-вартісного аналізу (ABC – Activity Based Costing), підтримувані, наприклад, у ППП Design/IDEF, Easy ABC+, ARIS ABC та ін. Так, функціонально-вартісний аналіз дозволяє виявити:

- найбільш трудомісткі й витратні функції;
- функції, що не роблять внеску в утворення прибутку;
- функції з низьким коефіцієнтом використання ресурсів.

Для динамічної оцінки наявних бізнес-процесів за наявності розвинутої функціональної інформаційної системи й інформаційного сховища можуть використовуватися методи аналізу реальної статистики.

У протилежному випадку застосовуються методи і засоби імітаційного моделювання, наприклад, ППП ReThink, РДО, Pilgrim, Ithink, Workflow Analyser, Service Model та ін.

3. Прямий інжиніринг.

Фаза прямого інжинірингу включає два етапи – розробки моделей нової організації бізнес-процесів і реалізації проекту реінжинірингу бізнес-процесів.

Розробка моделей нової організації бізнес-процесів передбачає, що моделі можуть будуватися в декількох варіантах, з яких вибираються найбільш ефективні варіанти з погляду реалізації критичних факторів успіху. У кінцевому рахунку залишають два варіанти моделей нової організації бізнес-процесів:

ідеальну модель, що може бути досягнута в перспективі і до якої варто прагнути;

реальну модель, що може бути досягнута в доступному для огляду майбутньому з урахуванням наявних ресурсів і обмежень. Причому реальна модель бізнес-процесів повинна бути такою, щоб можна було в перспективі перейти до ідеальної моделі, тобто необхідно, щоб обидві моделі були концептуально близькі одна одній.

Моделювання предметної області бізнес-процесів виконується в два підетапи:

визначається структура нової організації бізнес-процесів (склад об'єктів, функцій і подій);

об'єкти і функції бізнес-процесів розподіляються за структурними підрозділами підприємства і визначаються вимоги до інформаційної системи.

Для побудови архітектури інформаційної системи на цьому етапі формуються вимоги до того:

які функції будуть автоматизуватися;

які для цього будуть потрібні куповані й оригінальні типи програм;

які необхідні інформаційні об'єкти;

як вони будуть організовані й розподілені.

На фазі прямого інжинірингу доцільно використовувати програмні засоби моделювання, що дозволяють відобразити результати моделювання предметної області в репозиторії, використовуваному на всіх наступних етапах розробки проекту. Під **репозиторієм** мається на увазі сховище метаінформації про проект. До таких програмних засобів належать або CASE-засоби, наприклад Oracle Designer 2000, SilverRun, Natural Engineering Workbench, Rational Rose та ін., або модельно-орієнтовані

засоби компонентного проектування, наприклад, Business Engineer (SAP), Enterprise Modeller (BAAN)), або спеціалізовані, які дозволяють експортувати моделі в CASE-засоби чи модельно-орієнтовані засоби компонентного проектування, наприклад ARIS Toolset.

Переваги спеціалізованих програмних засобів моделювання бізнес-процесів полягають у більшій орієнтованості на власне бізнес-реінжиніринг і передпроектний аналіз для інформаційної системи, у той час як CASE-засоби і модельно-орієнтовані засоби компонентного проектування – на розробку інформаційної системи. Зокрема, для спеціалізованих програмних засобів моделювання бізнес-процесів (наприклад, ARIS Toolset, Design/IDEF) характерна підтримка інструментів статичного функціонально-вартісного аналізу і динамічного імітаційного моделювання для оцінювання заданих критеріїв ефективності бізнес-процесів.

Для обґрунтування варіанта організації бізнес-процесів на етапі прямого інжинірингу широко використовуються методи динамічного імітаційного моделювання. Динамічний аналіз передбачає розгляд у часі множини одночасно виконуваних бізнес-процесів, у той час як статичний аналіз досліджує виконання одного бізнес-процесу поза зв'язком із зайнятістю ресурсів в інших процесах. Імітаційне моделювання бізнес-процесів дозволяє формувати такі характеристики, як продуктивність (обсяги виробництва, збуту, закупівель), часові й вартісні характеристики процесів і окремих функцій, ступінь і вартість використання ресурсів. Актуальність застосування методів динамічного аналізу в бізнес-реінжинірингу обумовлена необхідністю скороченням міжопераційних затримок, пов'язаних із використанням ресурсів у множині процесів.

Цілями проведення імітаційного моделювання бізнес-процесів можуть бути:

- 1) порівняння середніх значень і дисперсій динамічних характеристик різних альтернатив процесів за однакових вхідних даних (один сценарій моделювання на кілька моделей);

- 2) відшукування оптимальних значень змінних на певній множині можливих значень (кілька сценаріїв моделювання на одну модель);

- 3) визначення залежностей між різними факторами процесів у результаті спеціального аналізу отриманої статистики.

Розвинуті засоби імітаційного моделювання, такі, наприклад, як ReThink (Gensym), дозволяють:

підвищити ступінь обґрунтованості проектів щодо реорганізації діяль-

ності підприємства з урахуванням аналізу і прогнозування зовнішніх та внутрішніх факторів розвитку економічної ситуації;

аналізувати і прогнозувати діяльність підприємства з урахуванням множини варіантів організації бізнесу і різних схем поведінки підприємства на ринку;

оптимізувати використання матеріальних, фінансових, людських та інформаційних ресурсів на різних стадіях життєвого циклу проекту реорганізації підприємства;

розробляти обґрунтовані рекомендації щодо зміни організаційної структури підприємства і впровадження інформаційних технологій.

Після розробки моделей нової організації бізнес-процесів здійснюється **реалізація проекту** реінжинірингу бізнес-процесів, яка полягає в розробці забезпечувальних підсистем, які підтримують функціонування підприємства в нових умовах.

Для зміни структури організаційно-економічної системи здійснюються:

розробка організаційно-штатної структури підприємства;

розробка посадових інструкцій;

розробка системи стимулювання працівників;

навчання персоналу;

підготовка робочої документації.

Для створення нової інформаційної системи здійснюються:

генерація, настроювання, програмування і налагодження програмних модулів;

розробка і наповнення бази даних;

встановлення обчислювального устаткування і системи телекомунікації.

Для швидкої розробки інформаційної системи широко використовуються CASE-засоби автоматизації проектування або засоби конфігурації комплексних систем управління ресурсами підприємства (ERP-системи), наприклад R/3, BAAN IV, Oracle Application, "Галактика", БОСС та ін. Утім в іншому випадку для розробки оригінального програмного забезпечення можуть знадобитися засоби швидкої розробки додатків (RAD-технологія) і мови програмування 4-го покоління (4GL), наприклад ABAP4, JAM та ін.

Етап реалізації проекту реінжинірингу бізнес-процесів закінчується складанням відповідно до прийнятих стандартів проектної документації або за допомогою CASE-засобів, або спеціалізованих програмних засобів. У будь-якому випадку повинне забезпечуватися якісне оформлення проектної документації з включенням необхідних графічних ілюстрацій. При цьому може виконуватися автоматизований контроль на відповідність розділів проектної документації, забезпечуватися її своєчасне відновлення.

4. Впровадження проекту реінжинірингу бізнес-процесів.

Впровадження проекту РБП, як правило, здійснюється поетапно відповідно до пріоритетів, установлених на етапі ідентифікації бізнес-процесів. Велике значення на етапі впровадження приділяється комплексному тестуванню компонентів проекту, для чого використовуються спеціальні програмні засоби.

Впровадження проекту РБП передбачає його здачу приймальній комісії, у котру входять представники осіб, що ухвалюють рішення, і майбутні менеджери процесів. Перед звітом команди РБП на комісії можлива організація незалежної експертизи проекту з боку спеціально підібраної інспекційної групи.

Після впровадження спроектованих бізнес-процесів у реальну практику дуже важливо організувати аналіз досягнення заданих на початку реінжинірингу критеріїв (метрик) ефективності функціонування підприємства (benchmarking), на основі яких можна вчасно ухвалювати рішення про необхідність адаптації бізнес-процесів до зовнішнього середовища, що змінюється.

12.4. Моделювання бізнес-процесів

12.4.1. Інструментальні засоби моделювання бізнес-процесів

Успішне впровадження процесного підходу – непросте завдання. У цьому контексті принципово важливо використовувати професійні інструментальні засоби, що дозволяють описувати і проводити аналіз бізнес-процесів, робити їх прозорішими і керованими. У разі обирання такого засобу потрібно враховувати наступні аспекти:

- мету проекту;

- вимоги до інформації, що характеризує бізнес-процес і необхідної для аналізу й ухвалення рішень у рамках проекту;

- можливості інструментального засобу задовольнити вимоги до інформації;

- можливості інструментальних засобів щодо документування процесів.

Найбільш поширеними інструментальними засобами для моделювання бізнес-процесів на сьогодні є BPWin і ARIS Toolset. Перший використовується для невеликих за масштабами проектів: малих і середніх підприємств, невеликої групи розробників, невеликої тривалості проекту. ARIS Toolset, використовується для масштабних і тривалих проектів.

Бізнес-модель – це не просто комплект документів, що описує тільки бізнес-процеси підприємства. Насправді в основі бізнес-моделі завжди лежать бізнес-цілі підприємства, що визначають склад усіх базових компонентів бізнес-моделі:

бізнес-функції, які описують, *що* робить бізнес;

бізнес-процеси, які описують, *як* підприємство виконує свої бізнес-функції;

організаційна структура, яка визначає, *де* виконуються бізнес-функції та бізнес-процеси;

фази, що визначають, *коли* (у якій послідовності) повинні бути упроваджені ті чи інші бізнес-функції;

ролі, що визначають, *хто* виконує бізнес-процеси;

правила, що визначають зв'язок між *що, як, де, коли і хто*.

На сучасному ринку комп'ютерних засобів моделювання представлені різноманітні засоби, наприклад Design/IDEF, BPWin, ERWin, OOWin, ARIS Toolset, Rational Rose тощо. Серед цих засобів слід виділити насамперед систему ARIS. Система ARIS, розроблена німецькими фахівцями компанії IDS Sheer AG під керівництвом професора Шевера. Вона має значну популярність за кордоном. Багато провідних компаній використовують ARIS для оптимізації свого бізнесу. У рейтингу Gartner Group система ARIS займає провідне місце. Слід зауважити, що Gartner, Inc. (NYSE: IT) – дослідницька і консалтингова компанія, світовий лідер у галузі інформаційних технологій. Вона є постійним партнером 60 тис. клієнтів в 10 тис. різних організаціях. За допомогою ресурсів Gartner Research, Gartner Consulting і Gartner Events компанія працює з кожним клієнтом над дослідженням, аналізом і впровадженням IT в бізнес. Компанія Gartner заснована в 1979 р., знаходиться в Стамфорді, Коннектикуті, США, і має 3,9 тис. партнерів, зокрема 1 200 дослідників-аналітиків і консультантів у 75 країнах.

Система ARIS є комплексом засобів аналізу і моделювання діяльності організацій, а також розробки автоматизованих інформаційних систем. У її основу покладена методологія, що ввібрала в себе особливості різних методів моделювання, які відбивають різні погляди на досліджувану систему. Одна й та ж модель може розроблятися з використанням декількох методологій, що дозволяє використовувати ARIS користувачам з різними теоретичними знаннями і налаштувати його на роботу з системами, що мають свою специфіку.

Методологія, використовувана ARIS, включає безліч різних методологій, інтегрованих у рамках системного підходу. Це дозволяє

говорити про єдину архітектуру даної методології. ARIS підтримує чотири типи моделей, що відбивають різні аспекти досліджуваної системи:

організаційні моделі, що представляють структуру системи – ієрархію організаційних підрозділів, посад і конкретних осіб, різноманіття зв'язків між ними, а також територіальну прив'язку структурних підрозділів;

функціональні моделі, що містять ієрархію цілей, які стоять перед апаратом управління, із сукупністю дерев функцій, необхідних для досягнення поставлених цілей;

інформаційні моделі, що відбивають структуру інформації, необхідної для реалізації всієї сукупності функцій системи;

моделі управління бізнес-процесами, що представляють комплексний погляд на реалізацію ділових процесів в рамках системи.

Графічно такий підхід може бути представлений домом моделей ARIS (рис. 50):

У методології ARIS модель процесів через сумісний репозиторій зв'язується з організаційною, функціональною і інформаційною моделями, кожна з яких відображає свій аспект діяльності підприємства. Це дозволяє здійснити комплексний всебічний опис організації, що є складною соціально-технічною системою.

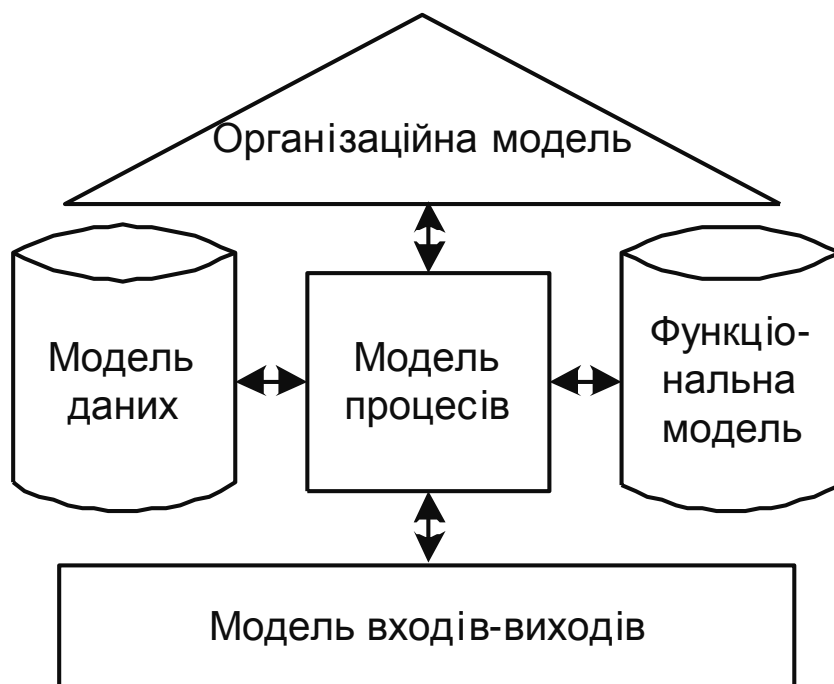


Рис. 50. Дім моделей ARIS

ARIS дозволяє побудувати модель всієї організації на основі так званих референтних моделей (рис. 51). При цьому описуються ніби різні зрізи підприємства, кожне зі своїми властивостями й атрибутами:

- моделі для опису організаційної структури підприємства;
- моделі для опису повноважень і прав доступу;
- моделі для опису матеріальних потоків і устаткування;
- моделі для опису продуктів;
- моделі для опису технічної інфраструктури;
- моделі для опису інформаційних систем і даних;
- моделі стратегічного планування;
- моделі для опису окремих аспектів діяльності;
- моделі для опису процесів;
- моделі для управління знаннями.

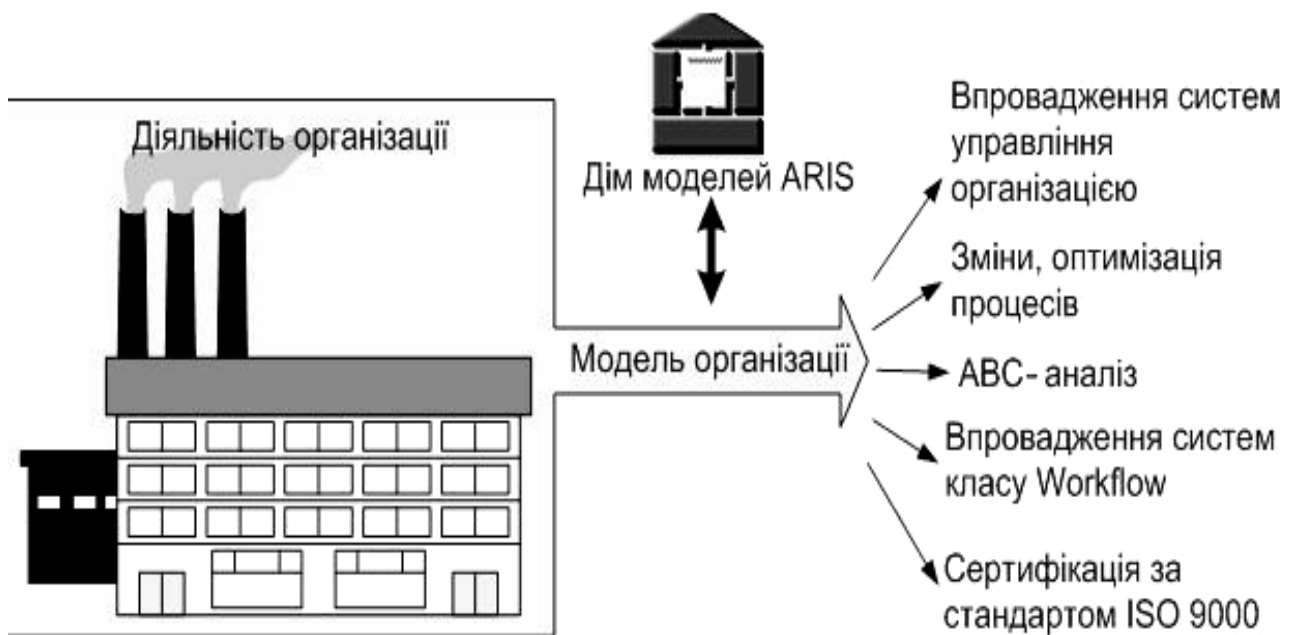


Рис. 51. Галузі застосування моделей ARIS

У сукупності виходить досить повна картина організації, що дозволяє проводити аналіз і реорганізацію бізнес-процесів (рис. 52). Реорганізація проводиться, як правило, після опису і моделювання існуючих процесів "як є" на основі результатів їх аналізу. Суть реорганізації полягає в проектуванні процесів "як повинно бути" або за рахунок радикальної перебудови існуючих процесів, або за рахунок підвищення їх якості.

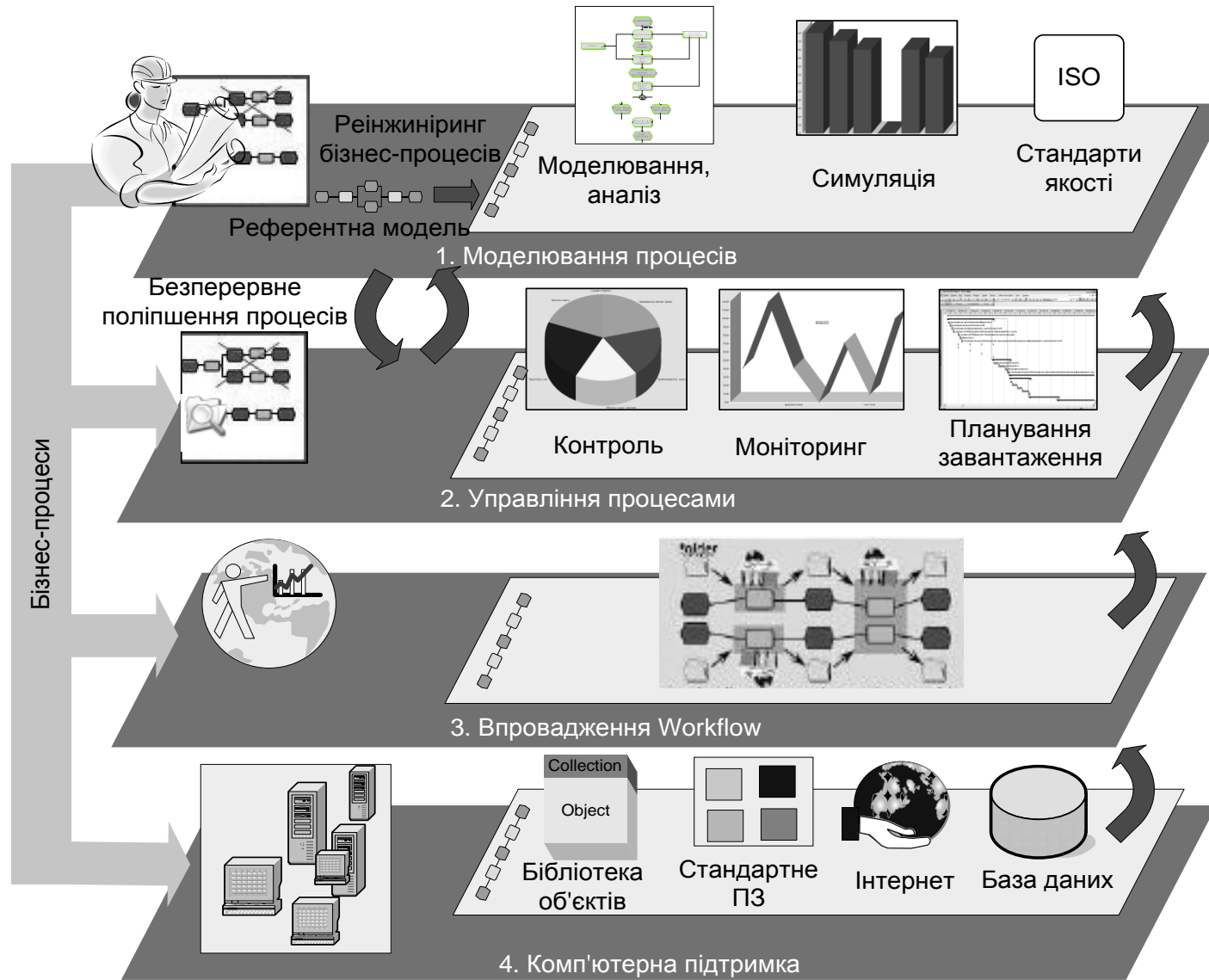


Рис. 52. Модель організації в методології ARIS

Найчастіше, перепроектування проводиться з урахуванням розробок процесно-орієнтованої системи менеджменту якості (СМЯ). Це передбачає розробку системи ключових показників продуктивності компанії, вироблення підходів для збирання інформації про реальні значення цих показників, щоб можна було ефективно управляти якістю процесів. Даний підхід забезпечить компанії основу для створення процесів, що відповідають світовим стандартам якості, і для сертифікації компанії за стандартами серії ISO 9001:2000.

Незважаючи на те, що для вирішення всього спектру завдань бажано мати повний набір програмних продуктів сімейства ARIS, можна вирішувати й окремі найбільш важливі завдання за допомогою обмеженого набору програмних продуктів. Нижче наведені найпопулярніші варіанти наборів компонентів програмних продуктів сімейства ARIS. Кожен із цих варіантів покликаний вирішити певні завдання і містить закінчений набір програмних компонентів, призначених для їх вирішення:

для опису і реінжинірингу бізнес-процесів проектною групою, розрахунку вартості виконання бізнес-процесів і моделювання їх роботи необхідне наступне: програмний продукт ARIS Toolset, ARIS Server і декілька програмних продуктів ARIS Easy Design, а також додаткові програмні модулі до ARIS Toolset – ARIS ABC і ARIS Simulation;

ARIS BSC призначений для швидкого моделювання стратегічної системи управління діяльністю компанії (Balanced ScoreCard). ARIS BSC забезпечує деталізацію та інтеграцію всіх стратегічних цілей, критичних чинників, ключових показників продуктивності компанії, а також їх аналіз і формування звітної документації;

ARIS Process Performance Manager автоматично збирає дані про продуктивність безпосередньо з бізнес-процесів компанії, зокрема дані з декількох різних інформаційних систем, роблячи таким чином можливим їх аналіз. Ця інформація може збиратися з інформаційних систем, наприклад, таких як ERP-, SCM- і CRM-системи, а також із систем класу workflow. ARIS Process Performance Manager забезпечує отримання інформації, що допомагає визначати "вузькі місця" в процесі виконання бізнес-процесів і оптимізувати час їх виконання;

ARIS Quality Management Scout – програмний продукт, що дозволяє створити систему, яка відповідає стандартам якості.

Відповідність стандартам високої якості – ключовий компонент довгострокового успішного управління компанією. Системи управління якістю створюють основу для інтегрованих систем управління і є необхідною вимогою досягнення успіху. Програмний продукт ARIS Quality Management Scout допоможе створити процесно-орієнтовану систему управління якістю, або допоможе узгодити існуючу систему управління якістю з вимогами стандартів ISO 9001:2000.

ARIS Process Risk Scout – це унікальний інструментальний засіб для управління операційними ризиками. Його компоненту Process Risk Assistant спрямовує користувача по кроках виконання всіх процесів, інструментальних засобів, інформації щодо ризиків і комунікацій у зв'язку з ризиками.

Таким чином, методологія ARIS дозволяє здійснити системний аналіз діяльності підприємства, створити його формалізований опис, провести аналіз одержаних моделей, спроектувати інформаційну систему і описати її реалізацію. У цій методології якнайповніше втілюються базові принципи системного підходу, що вимагає повного і всебічного розгляду досліджуваної системи. Тому на сьогодні ARIS є одним з найбільш перспективних інструментів аналізу і проектування складних соціально-технічних систем.

12.4.2. Опис бізнес-процесів у нотації ARIS

Нотація ARIS розроблена німецькими фахівцями компанії IDS Sheer AG під керівництвом професора Шеера. У нотації ARIS бізнес-процес подається таким чином (рис. 53).

Для моделювання бізнес-процесів використовується діаграма "Подієвий ланцюжок процесу". Подієвий ланцюжок процесу (модель або діаграма eEPC) призначений для детального опису, що виконується в рамках одного підрозділу, декількома підрозділами або конкретними співробітниками. В інструментальному середовищі ARIS Toolset ця модель дозволяє виявляти взаємозв'язки між організаційною і функціональною моделями. Для кожної функції можуть бути визначені початкова і кінцева події, відповідальні виконавці, матеріальні та документальні потоки, які супроводжують модель, а також проведена декомпозиція на нижчі рівні (підфункції і т. д.). Модель eEPC є найбільш інформативною і зручною під час опису діяльності підрозділів організації.

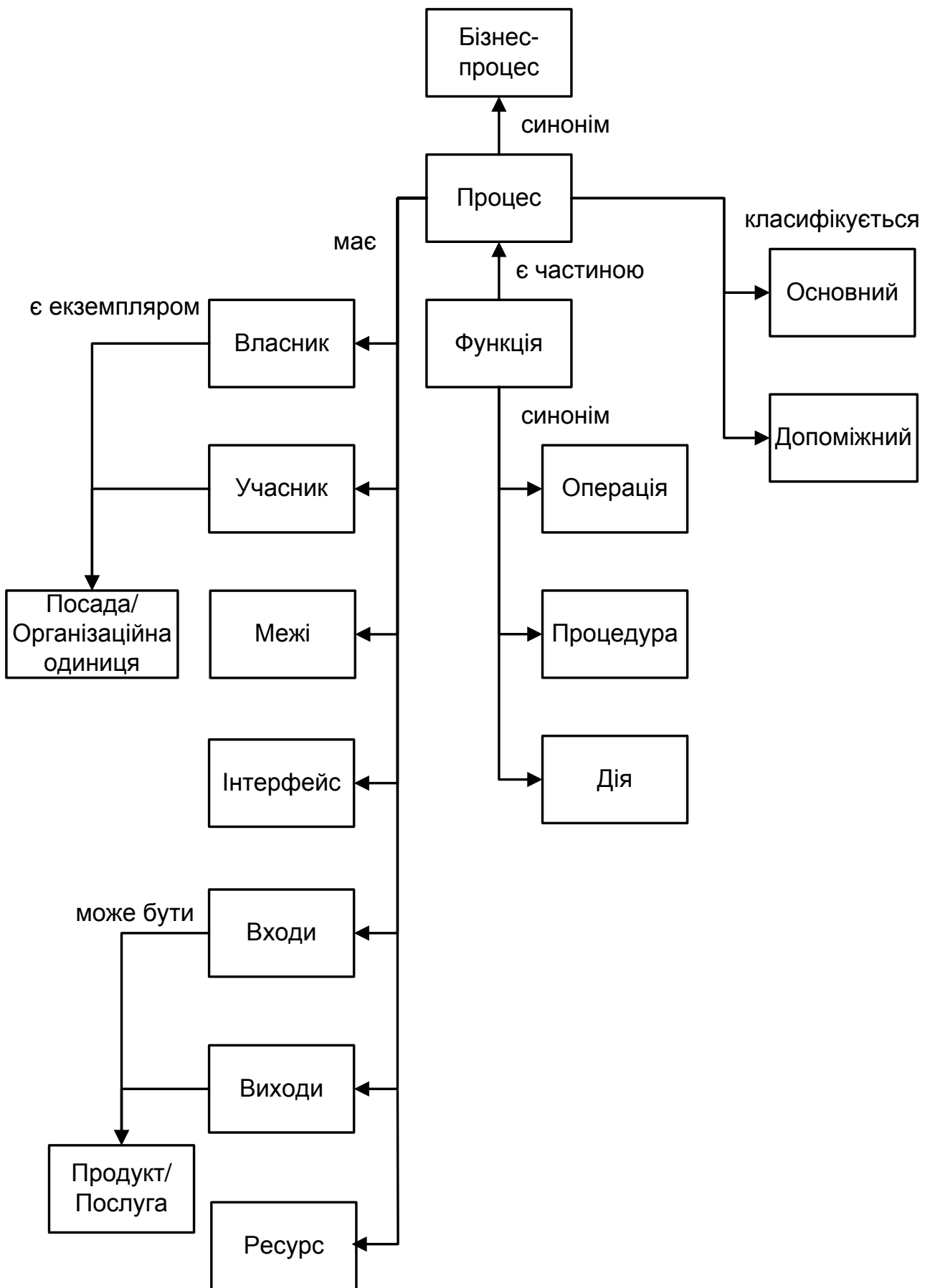


Рис. 53. Бізнес-процес у нотації ARIS

Проте можна виділити основні, найбільш часто використовувані об'єкти – це функція, подія, організаційна одиниця, документ, прикладна система, зв'язок, логічні оператори "І", "АБО", "АБО, що виключає" (табл. 24).

Особливістю моделі eEPC є те, що для кожної функції можуть бути визначені початкова і кінцева події, виконавці, матеріальні й інформаційні потоки. Подія – це стан, який є суттєвим для цілей управління бізнесом і який здійснює вплив або контролює подальший розвиток бізнес-процесів. Події активізують функції, тобто передають управління від однієї функції до іншої. Вони можуть бути також результатом виконання функцій. Події на противагу функціям не мають тривалості і здійснюються миттєво. За необхідності може бути проведена декомпозиція окремих функцій на підфункції.

Можна виділити наступні основні переваги нотації ARIS eEPC:

- добре сприйняття об'єктів моделі: кожен об'єкт відображається певним символом і має певний колір: функція – зелений, подія – рожевий, організаційна одиниця – жовтий і т. д.;

- використання різних типів зв'язків між об'єктами: сполучних і тих, що деталізують;

- використання подій і логічних операторів для відбиття логіки процесу бізнесу: кожна функція ініціюється подією і зароджує подію, яка ініціює виконання наступної функції;

- можливість відображення входів (процесів-постачальників) і виходів (процесів-клієнтів) процесу бізнесу;

- наочне відображення ресурсів процесу бізнесу: персоналу, устаткування, інформації, середовища виконання функції;

- можливість включення контрольних точок для вимірювання показників ефективності процесу;

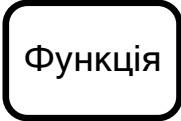
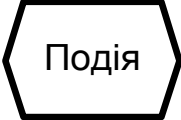
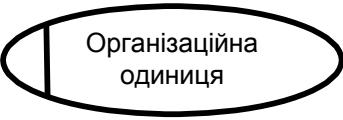

- можливість відображення відхилень від нормального процесу;

- наявність зв'язку виконавця і керівника процесу;

- зручність відображення моделі за допомогою її різновидів: у вигляді стовпчиків і рядків, у вигляді таблиць, з потоком матеріалів.

Процедурна послідовність функцій у рамках бізнес-процесів відображається у вигляді ланцюжка процесу, де для кожної функції можуть бути визначені початкова й кінцева події. **Подія** – це стан, який є суттєвим для цілей управління бізнесом і який впливає або контролює подальший розвиток одного або більше бізнес-процесів.

Основні об'єкти нотації ARIS eEPC

№	Найменування	Опис	Графічне позначення
1	Функція	Об'єкт "Функція" слугує для опису функцій (процедур, робіт), що виконуються підрозділами/співробітниками підприємства	
2	Подія	Об'єкт "Подія" слугує для опису реальних станів системи, які впливають і управляють виконанням функцій	
3	Організаційна одиниця	Об'єкт, який відображає різні організаційні ланки підприємства (наприклад, управління або відділ)	
4	Документ	Об'єкт, який відображає реальні носії інформації, наприклад паперовий документ	
5	Прикладна система	Об'єкт, який відображає реальну Прикладну систему, використовувану в рамках технології виконання функції	
6	Стрілка зв'язку між об'єктами	Об'єкт описує тип відносин між іншими об'єктами, наприклад – активацію виконання функції певною подією	
7	Логічне "І"	Логічний оператор, що визначає зв'язки між подіями і функціями в рамках процесу. Дозволяє описати розгалуження процесу	
8	Логічне "АБО"	Логічний оператор, що визначає зв'язки між подіями і функціями в рамках процесу. Дозволяє описати розгалуження процесу	
9	Логічне виключне "АБО"	Логічний оператор, що визначає зв'язки між подіями і функціями в рамках процесу. Дозволяє описати розгалуження процесу	

Події активізують функції, тобто передають управління від однієї функції до іншої. Вони можуть бути також результатом виконання функцій. На відміну від функцій, які мають певну тривалість, події відбуваються миттєво.

Опис події повинен містити не тільки інформаційний об'єкт (наприклад, "замовлення"), але й опис зміни стану ("отримане"). Події перемикають функції і можуть бути результатом виконання функції. Впорядкування комбінації подій і функцій у послідовність дозволяє створити подієві ланцюжки процесів.

На рис. 54 наведений фрагмент моделі процесу в нотації eEPC.

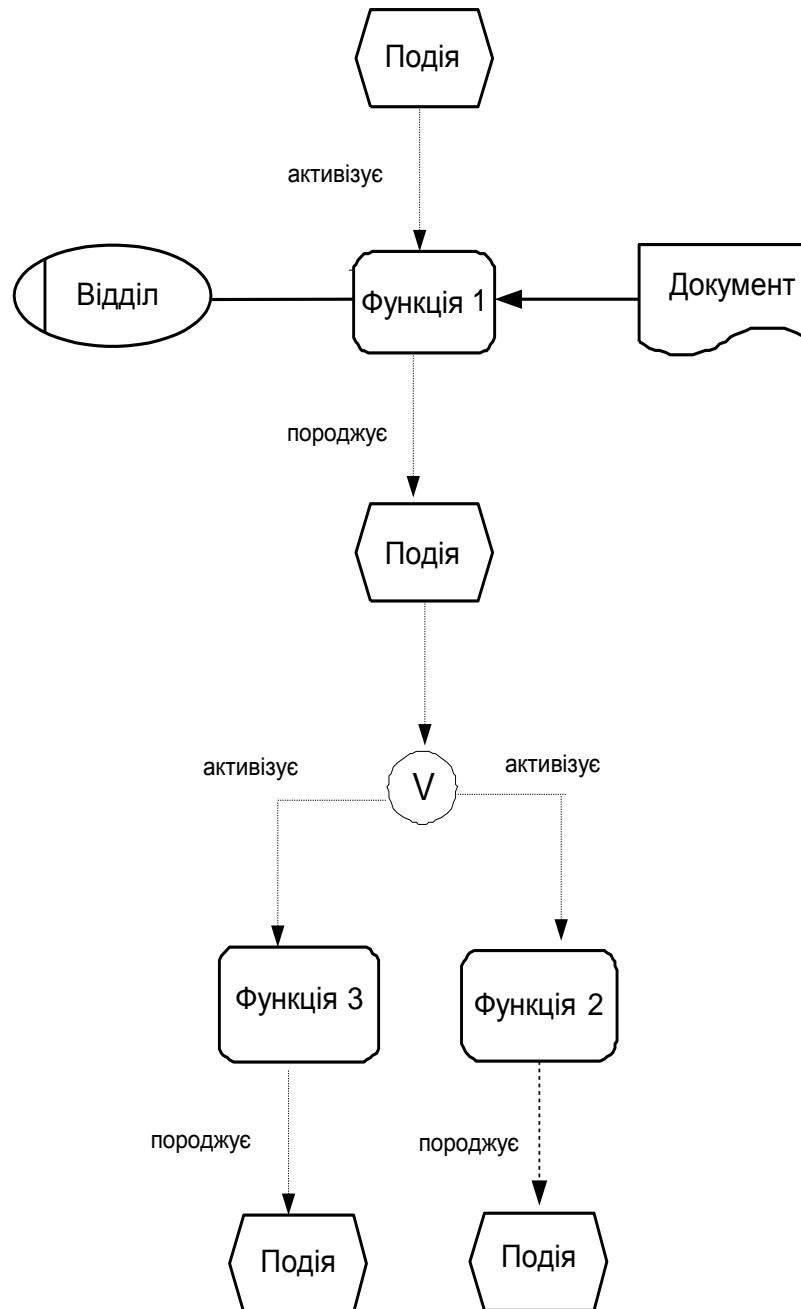


Рис. 54. Модель процесу в нотації eEPC

Розрізняють два типи операторів: оператори подій і оператори функцій. На рис. 55 наведені всі можливі оператори подій, на рис. 56 – функцій.

Особливу увагу необхідно приділити обмеженням, які існують для операторів функцій. Оскільки події не можуть ухвалювати рішення (тоді як функції можуть), подія, що перемикається, не повинна бути пов'язана операторами OR або XOR.

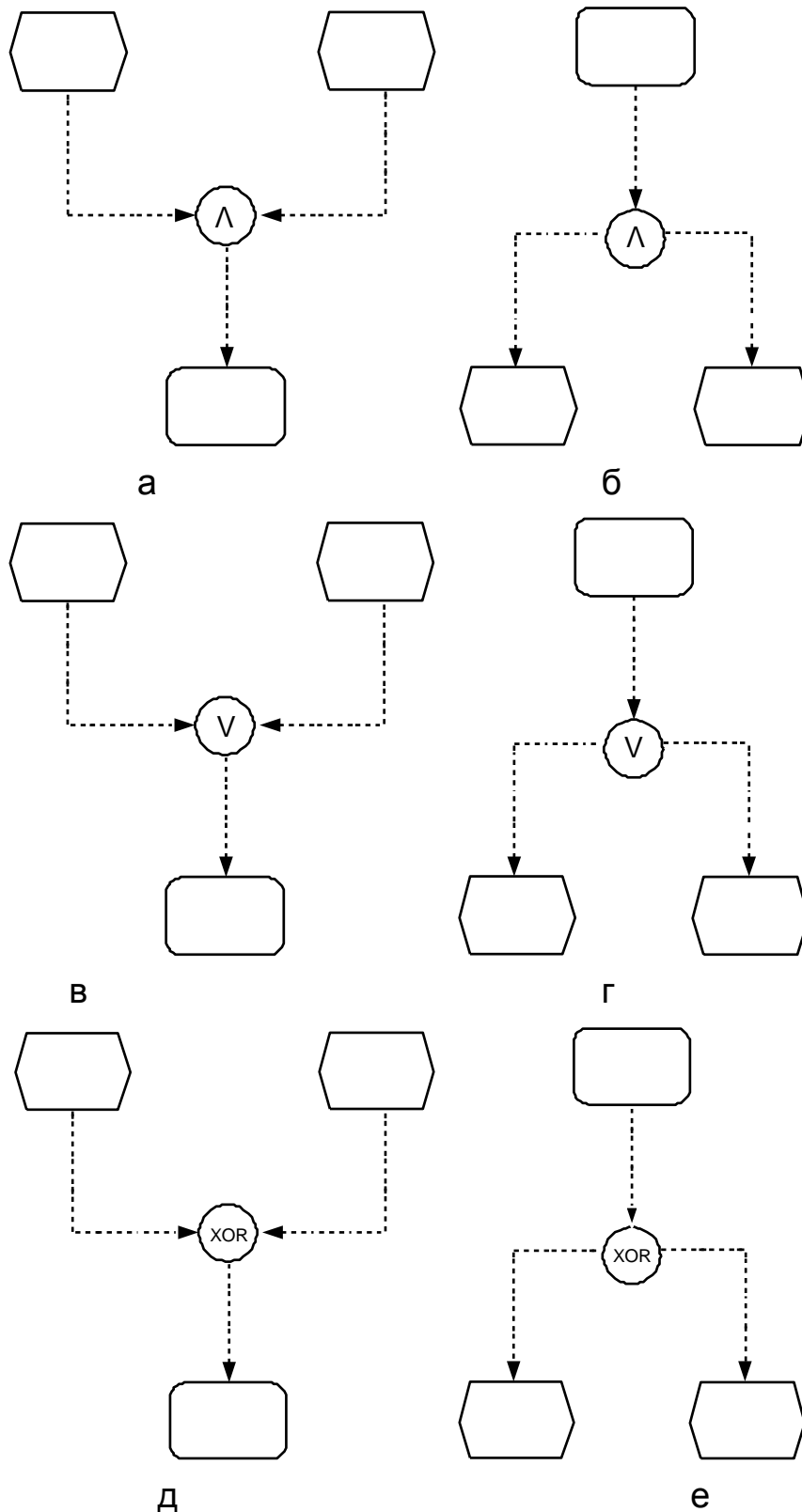


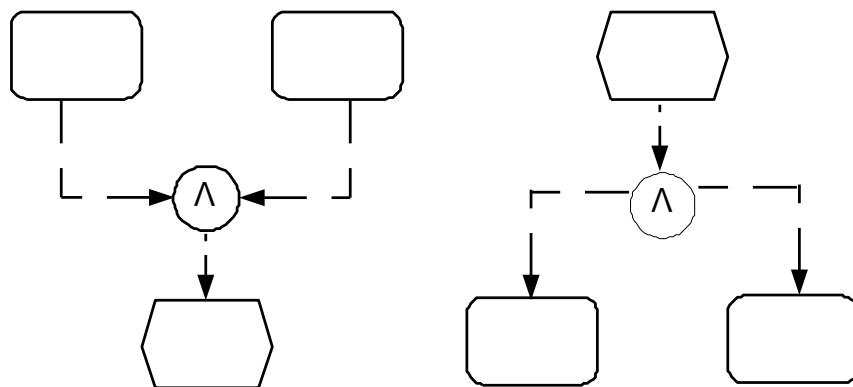
Рис. 55. Правила з'єднання подій

Умовні позначення:

а), б) з'єднання "І";

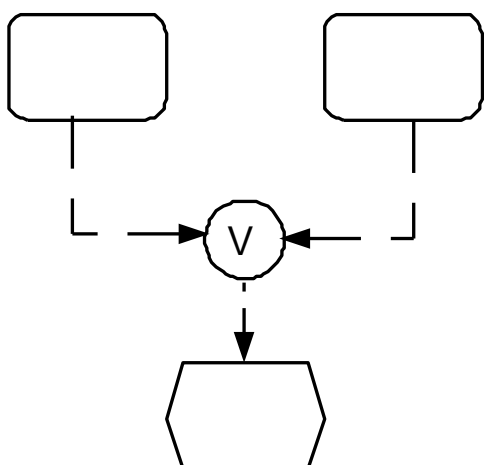
в), г) з'єднання "АБО";

д), е) з'єднання виключне "АБО".

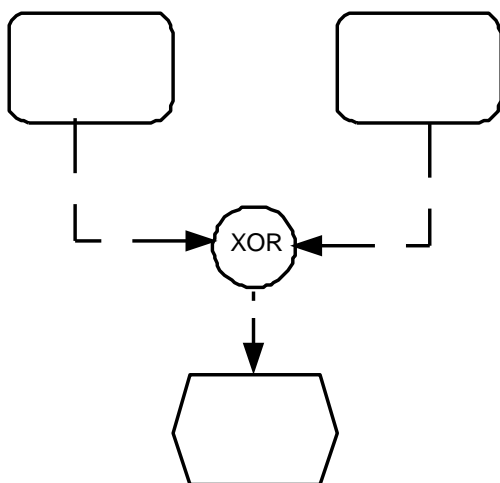


а

б



в



г

Рис. 56. Правила з'єднання функцій

Умовні позначення:

- а), б) з'єднання "І";
- в) з'єднання "АБО";
- г) з'єднання виключне "АБО".

Оператори OR і XOR не можуть використовуватися для з'єднання перемикальних подій і функцій, тому що події не можуть ухвалювати рішення. Оскільки функції викликаються послідовно, в моделі eEPC розгалуження і цикли обробки можуть виявитися представленими досить заплутаним способом.

Модель eEPC має ряд різновидів: у вигляді стовпчиків і рядків, у вигляді таблиць, з потоком матеріалів.

Від eEPC-моделі можна переходити надалі як до структурного, так і до об'єктного програмування системи. Моделі, створені за допомогою **ARIS Toolset**, є основою для реалізації ділових процедур у системах класу workflow (автоматизації ділових процесів) і процесів, побудованих за допомогою стандартного програмного забезпечення. Ці моделі слугують вхідною інформацією для CASE-систем, що використовуються для розробки нестандартного програмного забезпечення.

Контрольні запитання

1. Наведіть передумови виникнення комбінованого процесного підходу до проектування ІС.
2. Дайте визначення поняття "процес".
3. Що розуміють в процесному підході під деревом цілей, деревом функцій, деревом показників?
4. Які ключові ролі використовуються в процесному підході?
5. Дайте визначення понять, пов'язаних із межами процесу.
6. Дайте порівняльну характеристику ресурсів і входів процесу.
7. Наведіть схему управління процесом.
8. Порівняйте функціональний і процесний підходи до управління організаціями.
9. Що таке реінжиніринг бізнес-процесів, інжиніринг бізнес-процесів?
10. Охарактеризуйте фази реінжинірингу бізнес-процесів.
11. Які аспекти враховуються в процесі вибору інструментального засобу моделювання бізнес-процесів?

12. Дайте порівняльну характеристику інструментальних засобів моделювання бізнес-процесів BPWin і ARIS Toolset.
13. Яким чином представлений бізнес-процес в нотації ARIS?
14. Яке призначення і переваги моделі eEPC?
15. Які основні об'єкти використовуються в моделі eEPC?

13. Типове проектування ІС

13.1. Основні поняття і види методів типового проектування ІС

13.1.1. Види методів типового проектування

На сьогодні переважним є створення ІС шляхом адаптації типових програмних продуктів, що випускаються професійними виробниками. Методи типового проектування ІС передбачають створення системи з готових купованих типових елементів (типових проектних рішень). Для цього проектувана ІС повинна бути декомпозована на множину складових компонентів (підсистем, комплексів завдань, програмних модулів і т. д.), для яких підбираються і закупаються наявні на ринку типові проектні рішення. Як правило, вони включають програмні продукти. Далі куплені типові елементи настроюються на особливості конкретного підприємства чи допрацьовуються відповідно до вимог предметної області.

Під типовим проектним рішенням (ТПР) слід розуміти подане у вигляді проектної документації, включаючи програмні модулі, проектне рішення, придатне для багаторазового використання. Типові проектні рішення також називають **продуктами, що тиражуються**. Як проектне рішення може виступати реалізація окремих компонентів ІС (програмних модулів, функціональних завдань, автоматизованих робочих місць, локальних баз даних, локальних обчислювальних мереж) або взаємозалежних комплексів компонентів (функціональних і забезпечувальних підсистем у цілому).

Залежно від рівня декомпозиції системи розрізняють елементний, підсистемний і об'єктний методи типового проектування (рис. 57).

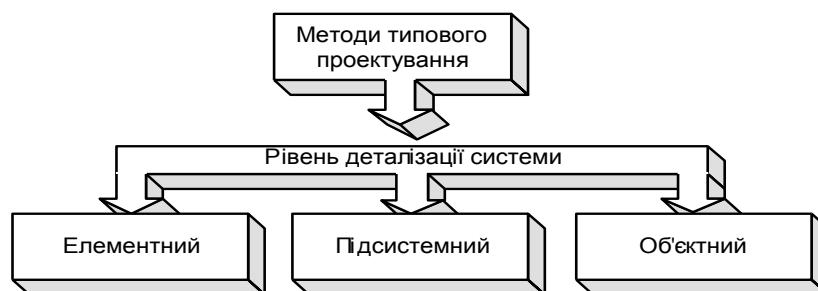


Рис. 57. Методи типового проектування

Порівняльна характеристика варіантів створення і розвитку ІС силами організації-користувача і спеціалізованими фірмами-розробниками програмних систем наведена в табл. 25.

Таблиця 25

Аналіз варіантів створення і розвитку ІС

Фази створення і функціонування ІС	Особливості засобів забезпечення ІС	
	власними силами	спеціалізованими фірмами
1	2	3
Проектування	Проект як такий може взагалі не випускатися, його роль зазвичай відіграють робочі матеріали програмістів; до початку робіт над системою, як правило, ніякої проектної документації на підприємстві немає	Випускається і постачається замовникові проектна документація в повному обсязі відповідно до специфікацій, вимог стандартів і традицій фірми; може бути розглянута в процесі укладання договору на постачання системи
Випробування	Створення спеціальних випробувальних засобів навряд чи буде здійснене у відчутному обсязі; скоріше за все, це будуть мінімальні можливості, які мають у своєму розпорядженні програмісти через якісь випадкові чинники	Спеціалізована фірма поступово створює розвинену базу для різноманітних випробувань своїх продуктів, оскільки це дозволяє підвищувати і гарантувати їх якість; вона забезпечує споживача набором відповідних засобів
Виготовлення	Один-єдиний екземпляр виготовляється під час проектування, тобто кустарно, на недосконалій технологічній базі	Випуск модулів ІС на основі наявних технологічних засобів, звичайно серійний, зі всіма вимогами до якості
Впровадження	Просто встановлення технічних засобів і програм на робочих місцях, у кращому разі – за певної участі майбутніх користувачів; оформлення акту здачі-приймання також не завжди має місце	Готові модулі системи планомірно встановлюються у споживача спеціалізованою бригадою, яка демонструє як власне цілковитий виріб, так і всі засоби його забезпечення
Освоєння	Навчання і консультації користувача здійснюють програмісти, для яких ця робота не є основною і привабливою	Виведення системи на проектну потужність або продуктивність за участю персоналу споживача здійснюється шляхом реалізації наперед відпрацьованої послідовності заходів, як технологічних, так і кадрових

1	2	3
Обслуговування	Спеціальні засоби для регулярного обслуговування практично не створюються; якусь роль можуть виконувати "домашні заготовки" програмістів, якщо вони мають досвід роботи в даній сфері	Виріб забезпечується спеціальними засобами обслуговування для проведення регулярних заходів щодо забезпечення працездатності, які фірма цілеспрямовано створює і відпрацьовує; вони входять у комплект постачання
Супровід	Роботи з підтримки працездатності елементів ІС і системи в цілому виконують програмісти, не маючи спеціалізованих засобів	Зазвичай формується спеціальна служба для роботи зі споживачами (відповіді на питання, попередження нарікань і т. д.); у комплект постачання ІС включаються спеціальні "фірмові" засоби та інструкції для проведення робіт із супроводу
Підтримка	Підтримку системи на підприємстві можуть здійснювати в основному програмісти-розробники, спираючись на свій і чужий досвід; звільнення програміста-розробника з підприємства в цих умовах може обернутися для ІС катастрофою	Фірма зацікавлена в збереженні клієнта, тому вона своєчасно сповіщає його про напрями розвитку системи, про ті можливості, які чекають клієнта надалі, а також про помічені недоробки та помилки і шляхи їх подолання; звільнення програмістів з фірми клієнт може навіть і не помітити

Типові проектні рішення створюються компаніями, що спеціалізуються на розробці програмних систем. Тому використання таких систем має значні переваги над системами, що створюються в організаціях власними силами.

13.1.2. Елементний метод типового проектування

Під час застосування елементного методу типового проектування ІС як типовий елемент системи використовується типове рішення щодо завдання чи окремого виду забезпечення завдання (інформаційного, програмного, технічного, математичного, організаційного) (рис. 58).



Рис. 58. Склад ТПР для завдання

Сутність застосування ТПР за елементного методу полягає в комплектації ІС з множини ТПР за окремими розрізненими завданнями. Якщо даної множини недостатньо для того, щоб спроектувати систему, необхідні модулі доопрацьовуються вручну. Переваги елементного методу типового проектування ІС пов'язані із застосуванням модульного підходу до проектування і документування ІС.

До недоліків застосування методу відносяться:

великі витрати часу на поєднання різнорідних елементів унаслідок інформаційної, програмної та технічної несумісності ТПР;

погана адаптивність (настроєність) елементів до особливостей підприємства.

Наслідком перерахованих недоліків є великі витрати часу на доробку і комплектування ТПР окремих елементів, порівняно з часом ручного оригінального проектування ІС. На сьогодні елементні ТПР в основному застосовуються як бібліотеки методо-орієнтованих програм (бібліотек класів об'єктів), наприклад, під час розробки графічних інтерфейсів, застосуванні обчислювальних і службових функцій.

13.1.3. Підсистемний метод типового проектування

У процесі використання підсистемного методу типового проектування ІС як елементи типізації виступають окремі підсистеми, що забезпечують функціональну повноту, мінімізацію зовнішніх інформаційних зв'язків, параметричну настроєність, альтернативність схем у межах значень вхідних параметрів. При цьому досягається більш високий ступінь інтеграції типових елементів ІС.

Типові проектні рішення для функціональних підсистем реалізуються у вигляді пакетів прикладних програм (ППП), що дозволяють здійснювати:

модульне проектування;

параметричне настроювання програмних компонентів на різні об'єкти управління;

скорочення витрат на проектування і програмування взаємозалежних компонентів;

якісне документування відображуваних процесів обробки інформації.

Недоліки підсистемного методу типового проектування:

недостатня з позиції безупинного інжинірингу ділових процесів адаптивність типових проектних рішень у вигляді функціональних ППП;

проблеми в поєднанні ППП різних функціональних підсистем, особливо у випадку використання ППП декількох виробників програмного забезпечення, для яких, як правило, характерна їх інформаційна, програмна і технічна несумісність між собою при побудові єдиної, корпоративної ІС.

Як приклади широко розповсюджених функціональних ППП можна назвати: "ІС: Підприємство" (автоматизація бухгалтерського обліку, розрахунку заробітної плати, складського обліку), "Фоліо – Склад" (автоматизація складських операцій), Project Expert (бізнес-планування), ІНЕК (фінансовий аналіз) та ін.

13.1.4. Об'єктний метод типового проектування

До сучасних типових проектів висуваються наступні вимоги:

відкритість архітектури, що дозволяє встановлювати проекти на різних програмно-технічних платформах;

масштабованість, що допускає конфігурацію ІС для змінної кількості робочих місць;

конфігурованість, що дозволяє вибирати компоненти, які необхідні для конкретної предметної області, і параметрично настроюватися на особливості об'єкта управління.

Усім цим вимогам відповідає об'єктний метод типового проектування

ІС. При об'єктному методі як типовий елемент використовується типовий проект для об'єктів управління визначеної галузі, що включає повний набір функціональних і забезпечувальних підсистем.

Безсумнівна перевага об'єктного методу типового проектування ІС перед підсистемним методом полягає в поєднанні всіх компонентів за рахунок методологічної єдності й інформаційної, програмної і технічної сумісності компонентів.

Існують два підходи до реалізації об'єктного методу типового проектування: параметрично-орієнтований і об'єктно-орієнтований. Адаптивність об'єктного методу проектування залежить від використовуваного підходу.

При **параметричному** настроюванні типових інформаційних систем, таких, наприклад, як ППП "Галактика", "Парус", "БОСС" та інші, виникають проблеми узгодження типового проекту з конкретним об'єктом управління так само, як і в разі використання підсистемного підходу. Звичайним способом вирішення проблеми адаптації є зміна структури організаційної системи об'єкта впровадження відповідно до вимог типового проекту або істотна доробка типового проекту за допомогою спеціальних інструментальних засобів типової системи.

Особливість модельно-орієнтованого підходу до реалізації об'єктного методу типового проектування ІС полягає в настроюванні типового проекту на особливості об'єкта управління шляхом узгодження моделі предметної сфери з моделлю типової системи. Підтримка при цьому моделі предметної області в репозиторії системи зближає метод типового проектування з методом автоматизованого проектування як у частині більш точного визначення і модифікації вимог до інформаційної системи, так і в частині коректності параметричного настроювання й автоматизованої доробки проектних рішень. Модельно-орієнтований підхід реалізації об'єктного методу типового проектування ІС відомий завдяки застосуванню типових інформаційних систем R/3 (SAP) і BAAN IV (BAAN).

13.2. Параметрично-орієнтоване проектування ІС

13.2.1. Основні потоки і компоненти ППП

У процесі проектування ІС на основі параметричного настроювання на вхід ППП задаються параметри і початкова інформація, а виходом слугує результат роботи пакета. У ППП входять наступні блоки: функціонування, обробки параметрів, адаптації (рис. 59). Слід розглянути

взаємозв'язок основних потоків і компонентів пакета прикладних програм.

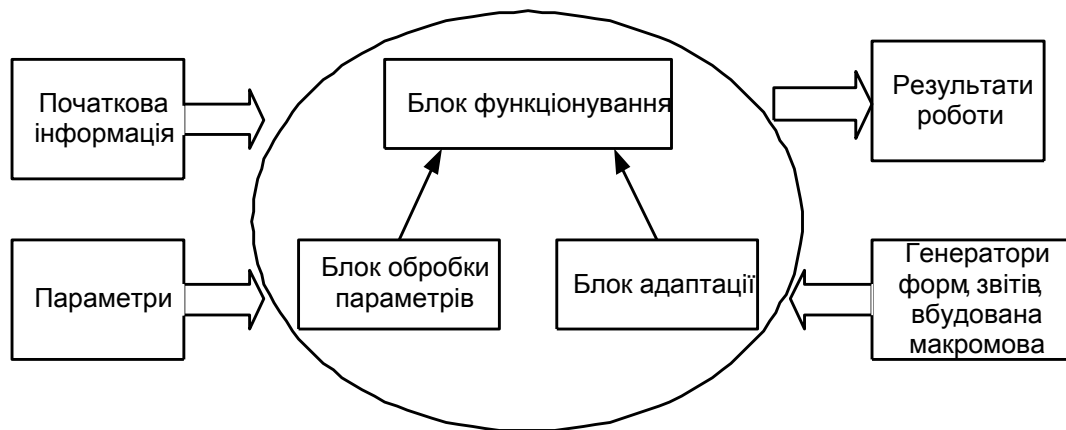


Рис. 59. Структура ППП за параметричного настроювання

Початкова інформація становить вхідні дані, що обробляються і необхідні для одержання результатів роботи пакета. Вхідні дані для функціонування пакета можуть бути подані у вигляді різних документів, причому як паперових, так і електронних.

Результати роботи пакета можуть бути подані у вигляді машинограм (звітів, таблиць, графіків), електронних документів, що можуть відображатися на екрані монітора, накопичуватися на технічних носіях чи спрямовуватися в зовнішнє середовище.

Блок функціонування обробляє вихідні дані і формує результати роботи пакета. Блок функціонування є деревом програмних модулів, що автоматизують функції обробки даних.

Параметри – це інформація, необхідна для настроювання пакета на конкретні умови функціонування. У параметри включають інформацію, що задається один раз під час установлення (інсталяції) цього пакета. Змінюючи параметри, можна додавати і вилучати які-небудь модулі чи впливати на режим їх роботи. У якості параметрів задаються користувачі і їхні рівні доступу до програмних модулів і до всього пакета в цілому.

Параметрична інформація подається:

у довідниках (класифікаторах з кількістю рівнів, що задається, класифікацією, наприклад, у довідниках номенклатури матеріалів, номенклатури виробів і послуг, одиниць виміру, валют і т. д.);

у таблицях описів конфігурації програмних модулів (наприклад, умови додавання (вилучення) модуля, режими ручного чи автоматичного відновлення полів даних, методи розрахунків показників і т. д.).

Блок обробки параметрів становить сукупність спеціальних модулів з інтерпретації значень параметрів. Зокрема, блок обробки параметрів переносить установки користувача безпосередньо в

прикладні програми й у використовувану базу даних. Проведене налаштування ППП дозволяє використовувати його для широкого класу об'єктів управління.

Блок адаптації взаємодіє з блоком функціонування і може додавати модулі або модифікувати їх. Необхідність застосування блоку адаптації пов'язана з потребами доробки програмних модулів ППП під впливом зовнішніх умов функціонування. Тому до складу ППП включається інструментарій адаптації наявних типових проектних рішень.

До таких інструментів, що доступні кінцевому користувачу – не програмісту, – застосовуються:

генератори програм ІС на основі мовних засобів RAD-технології (4GL);
макромови проектування і налаштування типових модулів.

13.2.2. Технологія проектування на основі параметричного налаштування ППП

У процесі типового проектування ІС на основі параметричного налаштування ППП виконуються наступні етапи:

визначення критеріїв оцінювання ППП;
оцінювання множини ППП-претендентів за сформульованими критеріями;

вибір і закупівля ППП із найвищою інтегральною оцінкою;
налаштування параметрів і можлива доробка закупленого ППП.

Слід розглянути докладніше їх зміст.

1. Визначення критеріїв оцінки функціонального ППП.

Перелік критеріїв вибору ППП для конкретної підсистеми визначається залежно від наступних характеристик предметної сфери: терміну розробки ІС, грошових ресурсів, технічної оснащеності об'єкта управління, ППП, що існують і функціонують, програмного і мережного оснащення та ін.

Можна виділити наступні групи критеріїв, що характеризують пакет прикладних програм:

призначення і можливості;
характерні ознаки і властивості;
вимоги до технічних і програмних засобів;
документація;
фінансові фактори;
встановлення;
експлуатація;
консалтинг;

оцінка якості та досвіду використання;
перспективи розвитку.

Критерії, які входять в зазначені групи, наведені в табл. 26.

Таблиця 26

Критерії оцінювання ППП

Група критеріїв	Критерій
1	2
1. Призначення і можливості	1.1. Предметна область використання
	1.2. Ступінь забезпечення функцій управління
	1.3. Загальний чи спеціалізований
	1.4. Колективного чи індивідуального користування
	1.5. Можливості розширення функцій
	1.6. Можливості оптимізації розрахунків
	1.7. Можливість адаптації до різних додатків на основі модульної будови
	1.8. Можливість взаємозамінності технічних засобів
	1.9. Можливість підвищення ефективності обробки даних
	1.10. Універсальність
	1.11. Застосовність для користувачів різної кваліфікації
2. Оптимальні ознаки і властивості	2.1. Вхідна мова
	2.2. Керівна мова
	2.3. Структура масиву
	2.4. Спосіб збереження даних
	2.5. Спосіб доступу до даних
	2.6. Видача вихідних документів
	2.7. Дистанційна обробка і поділ часу
	2.8. Подання вхідних даних
	2.9. Способи перевірки вхідних даних
	2.10. Подання вхідних даних
	2.11. Редагування вхідних даних
	2.12. Діалоговий режим
	2.13. Мова програмування
3. Вимоги до технічних програмних засобів	3.1. Обчислювальна система
	3.2. Обсяг ОП
	3.3. Обсяг зовнішньої пам'яті
	3.4. Периферійні пристрої
	3.5. Тип ОС
	3.6. Допоміжні програмні кошти
	3.7. Використання засобів організації масивів
	3.8. Сумісність із СУБД
4. Документація	4.1. Загальне керівництво щодо використання
	4.2. Керівництво системного й програмного рівня
5. Фінансові фактори	5.1. Витрати на придбання пакета
	5.2. Витрати на оренду пакета
	5.3. Витрати на встановлення й обробку пакета, підготовку персоналу, техніки, обслуговування й підтримку
	5.4. Економічна ефективність використання

1	2
6. Особливості встановлення	6.1. Обсяг робіт зі встановлення пакета 6.2. Час встановлення 6.3. Необхідні модифікації пакета 6.4. Вимоги користувача до кваліфікації програмістів 6.5. Вимоги настройки вхідних і вихідних форм 6.6. Трудомісткість організації інформаційної бази 6.7. Необхідні модифікації ОС і СУБД
7. Особливості експлуатації	7.1. Залежність робочих характеристик пакета від використовуваних технічних і програмних засобів 7.2. Можливість обслуговування фахівців організації-користувача 7.3. Технічна ефективність і надійність 7.4. Захист даних 7.5. Трудомісткість внесення змін 7.6. Трудомісткість реорганізації інформаційної бази 7.7. Трудомісткість і час виявлення та виправлення помилок 7.8. Час повторного запуску системи 7.9. Час циклу обробки інформації 7.10. Продуктивність
8. Консалтинг	8.1. Навчання персоналу організації-користувача 8.2. Участь постачальника в процесі впровадження пакета 8.3. Перехід від старої системи до нової 8.4. Участь постачальника в обстеженні пакета 8.5. Коректування системи помилок 8.6. Внесення модифікацій 8.7. Забезпечення відновлення документів 8.8. Простота використання
9. Оцінка якості та досвід використання	9.1. Джерело виникнення 9.2. Характеристика 1-ї версії пакета 9.3. Кількість і характер переробок пакета 9.4. Кількість організацій, які користуються пакетом 9.5. Оцінка вже встановлених пакетів 9.6. Порівняння з аналогічними пакетами 9.7. Перешкодозахищеність
10. Перспективи розвитку	10.1. Удосконалювання концепції й використовуваних методів 10.2. Підключення нових функціональних можливостей 10.3. Розширення інтерфейсу, перехід на досконалі технічні засоби

2. Оцінювання ринку функціональних ППП.

Оцінювання ринку функціональних ППП здійснюється серед програмних засобів, що є на ринку, на основі виділених груп критеріїв і може здійснюватися за методикою оцінювання ергономічних характеристик продуктів. За даною методикою передбачається усереднення оцінок групи експертів, що оцінюють ППП.

Для кожного критерію на основі оцінок декількох експертів за 10-бальною шкалою встановлюються середньозважені вагові коефіцієнти значимості, що нормуються всередині.

Для кожного ППП здійснюється експертна оцінка в розрізі окремих характеристик за 10-бальною шкалою. Далі оцінки корегуються на вагові коефіцієнти і нормуються всередині групи (формули 1 – 2):

$$K_{ji} = B_{ji} \times \omega_{ji}, \quad (1)$$

де K_{ji} – зважена бальна оцінка експертів i -го критерію j -ї групи;

B_{ji} – середнє значення бальної оцінки експертів для i -го критерію j -ї групи;

ω_{ji} – значення вагового коефіцієнта i -го критерію j -ї групи.

$$K_j = \sum_{i=1}^m K_{ji}, \quad (2)$$

де K_j – зважена бальна оцінка критеріїв j -ї групи.

Зважені бальні оцінки для груп критеріїв сумують в цілому за ППП (формула 3):

$$K = \sum_{j=1}^n K_j \cdot \omega_j, \quad (3)$$

де K – інтегральна оцінка в цілому для ППП.

ω_j – значення вагового коефіцієнта j -ї групи.

ППП, що одержав найбільшу зважену характеристику, є претендентом на ухвалення рішення про його придбання. У результаті ухвалення рішення про придбання ППП із фірмою-розробником укладається договір про постачання і супровід ППП разом із технічною документацією.

3. Настроювання функціонального ППП.

Настроювання ППП за технічною документацією починається із заповнення нормативно-довідкової інформації, необхідної для виконання функцій пакета. При цьому відбувається послідовне заповнення всіх необхідних довідників.

Далі відбувається настроювання модулів ППП, яке полягає в

параметризації функцій пакета. Як вхідну інформацію під час задавання значень тих чи інших параметрів використовують дані предметної області, а також технічну документацію пакета. Результатом настроювання модулів є ППП, готовий до експлуатації. Наприклад, настроювання модулів для підсистеми "Управління закупівлями" може зводиться до задавання наступної інформації:

системної інформації, що включає настроювання каталогів баз даних, прав користувачів, звітних періодів національної грошової одиниці і т. д.;

користувальницької інформації, що включає настроювання конкретних матеріально відповідальних осіб, складів, переліку документів, користувальницького інтерфейсу і т. д.

4. Навчання персоналу.

Навчання необхідне для ознайомлення, формування навичок використання ППП у персоналу, який буде безпосередньо працювати з ним. Під час навчання використовується технічна документація пакета. Результатом навчання персоналу є проходження різних контрольних заходів (тестів, іспитів і т. д.) і одержання документів, що свідчать про готовність персоналу до експлуатації ППП.

5. Експлуатація ППП.

Експлуатація ППП полягає в автоматизованому виконанні функцій управління за допомогою ППП. У процесі експлуатації ППП використовується вхідна інформація предметної сфери. При цьому збирається статистика роботи пакета, яка використовується для аналізу ефективності функціонування ППП і вироблення рекомендацій щодо його перенастроювання.

6. Адаптація типової конфігурації ППП до зовнішніх змін.

Під час виконання цього етапу використовують:

- 1) опис зовнішніх змін функціонування ППП;
- 2) технічну документацію ППП;
- 3) інструментальні засоби адаптації ППП.

Результатом виконання цього етапу є нова, адаптована версія ППП і оновлена технічна документація ППП.

У разі зміни умов функціонування використовуються наступні інструменти адаптації ППП:

генератори звітів, форм введення-виведення;
макромови настроювання функцій ППП;
вбудовані мови програмування.

Таким чином, параметрично-орієнтоване проектування ІС на основі використання ППП порівняно з оригінальним проектуванням дає

можливість більш швидкого і гнучкого впровадження інформаційної системи.

Однак існує ряд проблем, що стримують поширення даної технології. До них можна віднести наступні:

- психологічні й організаційні труднощі впровадження ППП;
- досить високу вартість придбання ППП і навчання персоналу;
- відсутність глобальної моделі об'єкта управління, що веде до витрат щодо узгодження різних ППП при проектуванні ІС.

13.3. Модельно-орієнтоване проектування ІС

13.3.1. Конфігурація ІС на основі модельно-орієнтованої технології проектування

Сутність модельно-орієнтованого проектування ІС зводиться до адаптації компонентів типової ІС відповідно до моделі предметної області конкретної організації. Для цього технологія проектування повинна підтримувати як модель типової ІС, так і модель конкретної організації, а також засоби підтримки відповідності між ними.

Ядром типової ІС є модель предметної області (організації), яка постійно розвивається і підтримується в спеціальній базі метаянформації – **репозиторії**, на основі якого здійснюється конфігурація програмного забезпечення. Таким чином, проектування й адаптація ІС зводяться насамперед до побудови моделі предметної сфери та її періодичного коректування.

Для моделювання предметної області і наступних конфігурацій інформаційної системи з окремих компонентів (програмних модулів) використовується спеціальний програмний інструментарій, наприклад SAP Business Engineering Workbench (BEW) або BAAN Enterprise Modeler. Безсумнівною перевагою застосування модельно-орієнтованих компонентних систем, таких як R/3 чи BAAN IV, перед CASE-технологіями є накопичення досвіду проектування інформаційних систем для різних галузей і типів виробництва у вигляді **типових моделей**, що постачаються разом із програмним продуктом у формі наповненого репозиторію. Таким чином, разом із програмним продуктом користувачі здобувають базу знань "know-how" про ефективні методи організації та управління бізнес-процесами, які можна адаптувати відповідно до

специфіки конкретного організаційного об'єкта.

Репозиторій корпоративної ІС, що використовує модельно-орієнтовану технологію проектування, у загальному випадку містить наступну метайнформацію:

базової моделі функціональності типової системи (посилальної моделі);
типових моделей визначених класів ІС (референтних моделей);
моделі організації, одержуваної на основі базової або типової моделей.

Базова модель містить опис бізнес-функцій, бізнес-процесів, бізнес-об'єктів, бізнес-правил, організаційної структури, що використовуються в програмних модулях типової ІС. При цьому велике значення в базовій моделі має задавання **бізнес-правил** підтримки цілісності інформаційної системи, що визначають умови перевірки коректності спільного застосування різних компонентів ІС. Таким чином, різноманіття і гнучкість визначення бізнес-процесів і відповідних конфігурацій інформаційної системи задаються за допомогою набору бізнес-правил.

Типові моделі описують конфігурації інформаційної системи для визначених галузей (автомобільної, електронної, нафтогазової та ін.) або типів виробництва (одиночного, серійного, масового, безупинного та ін.).

Модель організації (предметної області) будується або шляхом узгодження фрагментів базової чи типової моделі відповідно зі специфічними особливостями організації, наприклад як в інструментальному засобі BAAN Enterprise Modeller, або в результаті перегляду цих моделей і експертного опитування, як в інструментальному засобі SAP Business Engineering Workbench. Причому в останньому випадку користувачу пропонується визначити значення не всіх параметрів, а тільки тих, котрі пов'язані між собою в контексті діалогу й описані бізнес-правилами.

Побудована модель організації у вигляді **метаопису** зберігається в репозиторії і за необхідності може бути відкоректована. Далі за моделлю організації автоматично здійснюється конфігурація інформаційної системи, в ході якої виконується семантичний контроль за бізнес-правилами.

В узагальненому вигляді конфігурація інформаційних систем на основі модельно-орієнтованої технології наведена на рис. 60.

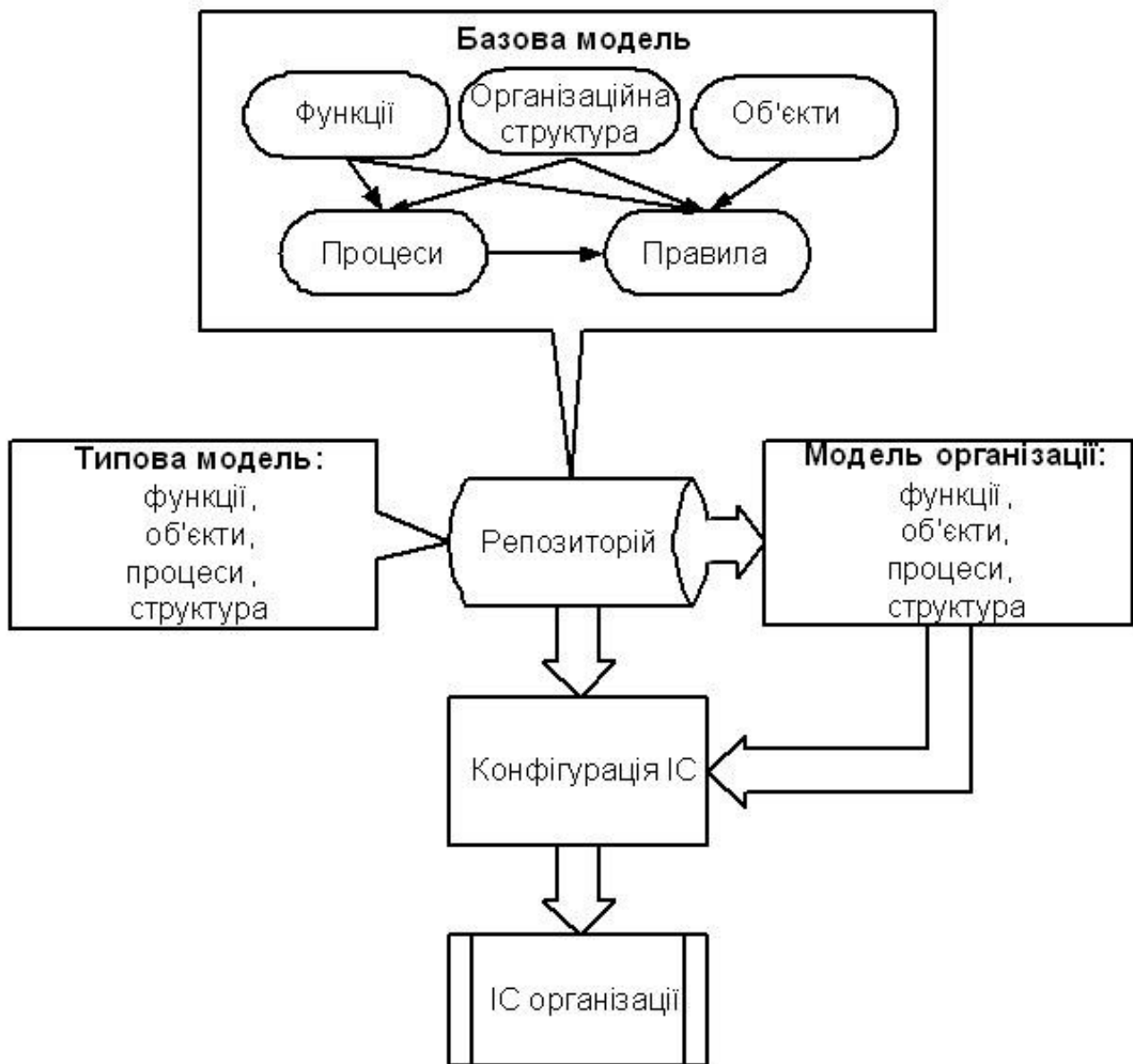


Рис. 60. Конфігурація ІС на основі модельно-орієнтованої технології

Далі слід розглянути більш докладно модель організації.

13.3.2. Компоненти моделі організації

У процесі моделювання предметної області організація подається у вигляді ієрархічної моделі, що формується з елементарних функціональних модулів (функцій), які з'єднуються в інформаційні потоки (процеси), і організаційної структури підприємства (організації). Сукупність моделей функцій, моделей процесів і моделі організаційної структури становить модель організації (рис. 61).

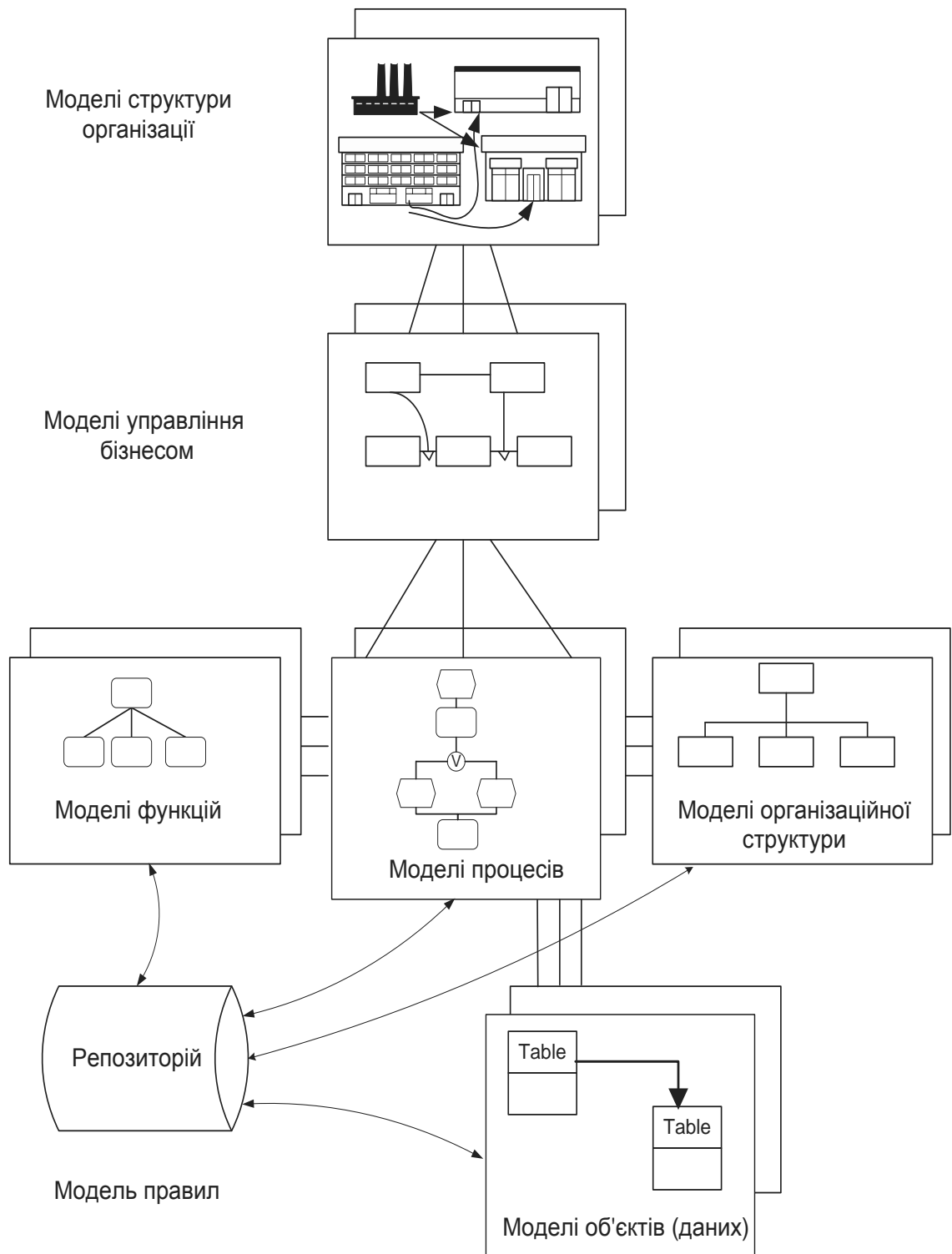


Рис. 61. Модель організації

Модель організації включає такі компоненти:
 модель функцій;
 модель процесів;
 модель організаційної структури;
 модель об'єктів (базу даних для визначеної галузі);
 модель правил.

Під час моделювання використовуються такі підходи:

діяльність організації розподіляється на види відповідно до функціонально-організаційної структури;

види діяльності пов'язані двома типами зв'язків – інформаційними і керівними. Перший передає необхідну вхідну інформацію для ініціації діяльності, а другий передає право дії (подія виконання, під час надходження якої на вхід починається виконання процесів, які становлять сутність діяльності);

якщо діяльність має більше однієї функції в процесі, управління переходами здійснюється в такий же спосіб – передачею вхідних даних і подій каналами, що зв'язує функції в процеси;

після виконання діяльності і за її результатами подія копіюється в один (чи декілька) відповідних каналів для передачі факту виконання діяльності;

після виконання діяльності, на основі її результатів (чи на основі сукупності результатів кількох видів діяльності) визначається один і тільки один новий напрямок передачі події.

13.3.3. Модель функцій

Модель функцій становить ієрархічну декомпозицію функціональної діяльності організації. До її складу входять:

дерево функцій;

відношення оптимізації;

"ноу-хау" в галузі вдосконалення бізнесу;

ключові показники діяльності.

На першому рівні ієрархії звичайно вказуються основні види функціональних підсистем. Наприклад, можна виділити виробництво, збут, логістику, сервіс, фінанси, персонал і т. д. На наступному рівні ієрархії для кожної функціональної підсистеми показуються функціональні модулі. Наприклад, для підсистеми "Логістика" виділяють функціональні модулі: планування потреби в матеріалах, закупівлі, управління запасами, управління складами, перевірки платежів і т. д. Для функціональних модулів задаються набори бізнес-функцій, для кожної з яких надалі визначаються бізнес-процеси. Наприклад, для функціонального модуля "закупівлі" визначаються бізнес-функції: оформлення договорів, оформлення замовлень, виписки рахунків і т. д.

Приклад фрагмента моделі бізнес-функцій показаний на рис. 62.

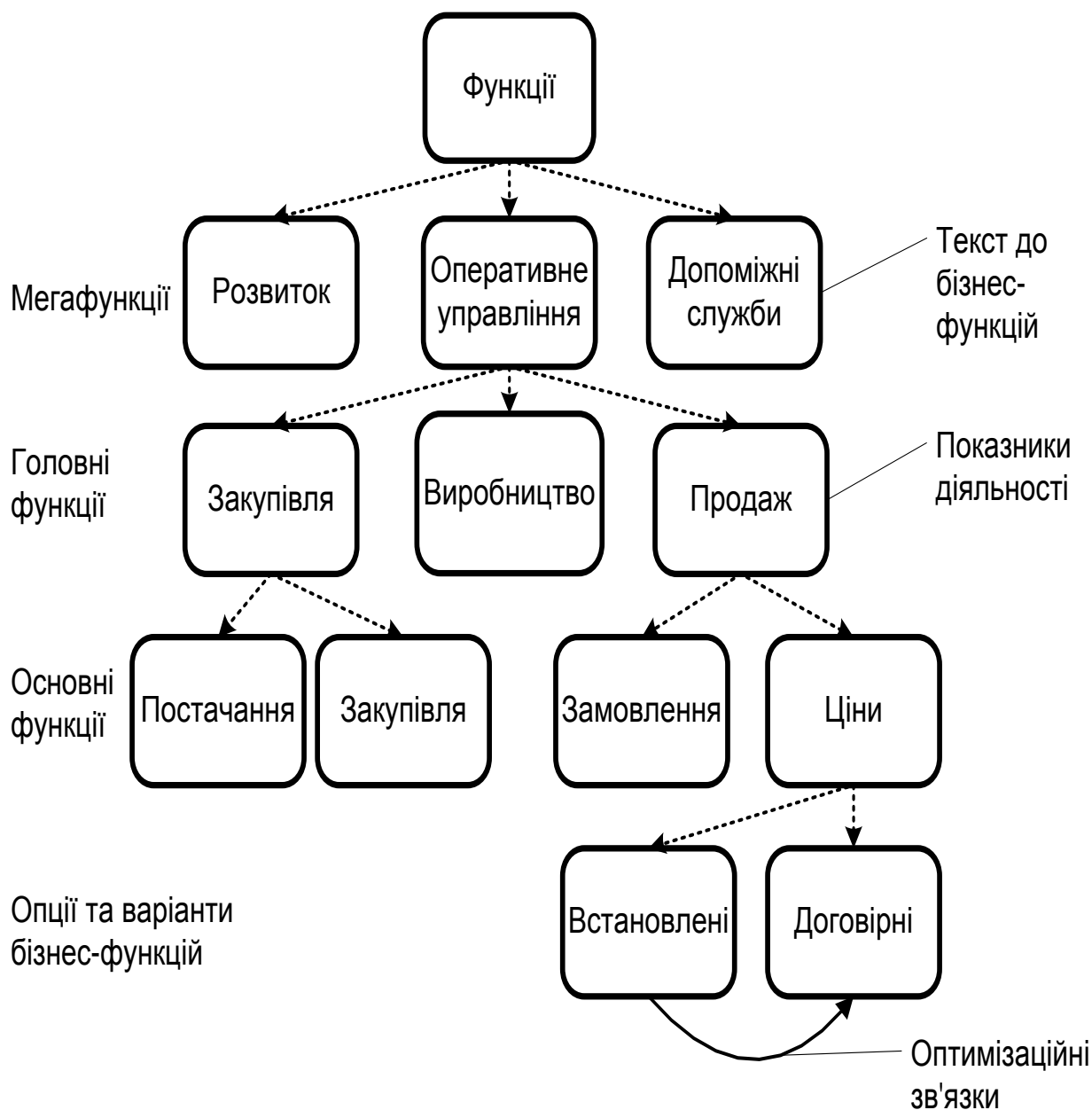


Рис. 62. Фрагмент моделі функцій

Бізнес-функції використовуються для досягнення конкретних цілей. У моделі функцій вони описуються термінами, звичайними для бізнесу (а не термінами корпоративної системи). На нижчому рівні бізнес-функції можуть варіюватися залежно від варіанта моделі підприємства. У цьому випадку подібні функції називають **варіантами бізнес-функції** ІС (наприклад, функція з інтерактивним чи автоматичним виконанням). Моделювання функцій призначене для досягнення таких цілей:

модель функцій може бути використана як перший логічний крок у процесі моделювання, в якому на базі загальних цілей і моделі функцій може бути створена модель процесу;

модель функцій може застосовуватися як допоміжний засіб для консультантів під час проведення презентацій для керівництва, на яких демонструється модель підприємства для визначеної галузі промисловості;

модель функцій може бути засобом подання галузевого "ноу-хау" у сфері вдосконалення господарської діяльності (наприклад, розширення напрямків оптимізації бізнес-функцій), найкращої практики ведення бізнесу і показників діяльності;

модель функцій може бути використана як інструмент перетворення нейтральної моделі підприємства (у термінах, звичних для бізнесу) у специфічну для корпоративної системи модель підприємства.

Основні характеристики моделі функцій наступні:

функції подаються у вигляді **вузлів** у деревоподібній діаграмі, які може переміщувати користувач;

функції можуть пояснюватися **текстами**, в яких описуються характеристики окремих функцій, пов'язуватися з документацією, що її описує, (наприклад, інструкції для виконання);

з функціями можуть бути пов'язані **показники діяльності** (показники оцінки ефективності їх виконання). У моделях підприємства конкретному показнику діяльності може бути приписане стандартне значення;

для функцій останнього рівня (бізнес-функцій) зазначаються **варіанти реалізації (оптимізації)** у міру впровадження. Відносини оптимізації можуть бути пов'язані з текстами, що описують організаційні аспекти реалізації процесів; а в моделі організації варіанти бізнес-функцій, об'єднані відносинами оптимізації, можуть бути пов'язані з етапами впровадження.

Архів узагальнених бізнес-функцій для всієї промисловості має потребу в системі класифікації, для того щоб мати можливість відшукати необхідну структуру в процесі формування референтної моделі. У системі моделювання функцій для класифікації розглядається ієрархічне дерево бізнес-функцій, що відповідає стандартній структурі. У межах цієї структури визначені чотири ієрархічних рівні: мегафункції, головні функції, основні функції, опції та варіанти бізнес-функцій.

Мегафункції. Кожна компанія може бути класифікована відповідно до шістьох мегафункцій (табл. 27). Ці мегафункції мають узагальнений характер і, отже, не підпорядковуються певній спеціалізації діяльності.

Згадані функції охоплюють і управління, і процеси виконання (стратегічне планування, розвиток і виконання).

Таблиця 27

Стандартна класифікація бізнес-функцій

Мегафункції	Головні функції
1. Придбання нового бізнесу	1.1. Дослідження ринку. 1.2. Перспективи, клієнтура, співвідношення. 1.3. Рекламування, публікація, зв'язок. 1.4. Технологічні нововведення, системні дослідження. 1.5...
2. Виконання (реалізація)	2.1. Ділове планування. 2.2. Контроль виконання. 2.3. Діловий контроль (управління). 2.4...
3. Підтримка	3.1. Фінансовий бухгалтерський облік. 3.2. Планування і контроль (управління). 3.3. Казначейське планування. 3.4. Робота з персоналом. 3.5...
4. Новий виріб і проект обслуговування	4.1. Нововведення виробу. 4.2. Нововведення процесу. 4.3. Нововведення обслуговування (служби). 4.4...
5. Дії	
5а. Продаж	5а.1. Замовлення і пропозиція. 5а.2. Комерційне опрацювання замовлення. 5а.3. Асигнування і фінансова оцінка. 5а.4 ...
5б. Розробка	5б.1. Інженерне проектування. 5б.2. Проведення розробки. 5б.3. Документація. 5б.4. ...
5с. Планування	5с.1. Проектування. 5с.2. Планування виробництва та управління. 5с.3. Планування інвентаря. 5с.4...
5д. Виробництво	5д.1. Управління цехом. 5д.2. Збір даних. 5д.3. Реєстрація використовуваних матеріалів. 5д.4. ...
5е. Доставка	5е.1. Приймальні іспити. 5е.2. Транспортування і встановлення. 5е.3....
5ф. Закупівля	5ф.1. Придбання матеріалів. 5ф.2. Купівля товарної продукції. 5ф.3. Висновок субпідрядних договорів. 5ф.4. ...
6. Післяпродажна підтримка	6.1. Обслуговування скарг. 6.2. Гарантійне обслуговування. 6.3. Післягарантійний ремонт. 6.4. Планування і контроль системи обслуговування

Головні функції. Кожна мегафункція згідно з функціональною

декомпозицією може включати декілька головних функцій. Головна функція тісніше пов'язана зі спеціалізацією бізнесу, ніж мегафункція. Тому архів головних функцій буде зростати в процесі збільшення кількості спеціалізованих моделей бізнесу.

Основні функції. Кожна основна функція подає головний процес у процесній моделі. У сукупності вони визначені для одного варіанта документопотоку.

Опції та варіанти бізнес-функцій. Основні функції виражаються через формування визначених підпорядкованих функцій. Крім цього, опції та варіанти можуть бути визначені обслуговочною системою для заміни чи доповнення основного змісту бізнес-процесів.

13.3.4. Модель процесів

Модель процесів – це опис загального виробничого процесу в термінах конкретної корпоративної системи. Така модель описує, які бізнес-процеси у визначеному напрямку діяльності можуть підтримуватися конкретною корпоративною системою, і яким чином це може бути здійснено.

Модель процесу відбиває послідовність виконання робіт (дій, операцій) для функцій найнижчого рівня моделі функцій, що дозволяє провести конфігурацію програмних модулів інформаційної системи відповідно до характерних рис конкретної предметної області.

У моделі можна створити ієрархію, в результаті чого, наприклад, потік замовлень всередині організації може бути змодельований у так звану "основну" процедуру, у той час як подробиці про даний потік замовлення рекомендується наводити в схемах детальних процедур.

Модель процесів призначена для таких цілей:

дати наочне подання того, яким чином корпоративна система працює з бізнес-процесами найвищого рівня, особливо для визначеного напрямку чи виду діяльності (основні процедури), наприклад, "Потік замовлень основної процедури в моделі комерційної діяльності";

дати наочне уявлення і задокументувати загальні бізнес-процеси (детальні процедури), що підтримуються системою (наприклад, "Надходження від продажу");

дати наочне уявлення і задокументувати періодичні детальні бізнес-процеси (наприклад, "Рознесення фінансових операцій" і "Підрахунок циклів");

дати наочне уявлення і задокументувати процедури початкового запуску і настановні процедури, необхідні для впровадження системи (наприклад, "Ввести модуль обробки повідомлень банку" або "Визначити точки фінансових операцій");

задокументувати адміністративні процеси, з якими можна зв'язати виконавців бізнес-процесів залежно від роду діяльності, посади, повноваження, посадових інструкцій тощо;

розробити і задокументувати перетворення корпоративною системою бізнес-процесів, пов'язаних із конкретним проектом.

Бізнес-процеси мають такі характеристики:

бізнес-процеси складаються з ряду компонентів, а саме подій, робіт і логічних операторів (і, або, виключне або). **Робота** є основною частиною бізнес-процесів. **Подія** передує кожному виду роботи і визначається цим видом роботи. Логічні оператори використовуються для управління бізнес-процесом;

у моделі процесу можуть бути показані: входи і постачальники, виходи і клієнти, ресурси, персонал, обладнання, інформація, середовище;

роботи бізнес-процесу, як і процес у цілому, документуються;

до робіт можуть бути прив'язані інформаційні й матеріальні одиниці, посадові інструкції, документи і прикладні програми;

працівники і виконувані ними функції можуть бути співвіднесені з видами робіт бізнес-процесу;

залежно від установлених значень атрибутів компонентів модель бізнес-процесу може бути симульована динамічно. У результаті із самого початку можна перевірити бізнес-процес для конкретних, заздалегідь заданих умов.

Бізнес-процеси розділяють на головні й деталізовані; основні та допоміжні.

Головні процеси, які відтворюють загальний товаро- і документообіг, що відповідає концептуальній моделі, визначені спеціалізацією бізнесу і, в принципі, є унікальними.

Деталізовані процеси мають узагальнюючий характер і можуть застосовуватися в багатьох секторах.

Основні процеси – це ті процеси, які додають нової якості продукції.

Допоміжні процеси формують інфраструктуру організації.

13.3.5. Модель правил

Для побудови бізнес-процесів з бізнес-функцій і об'єднання останніх у

рамках моделі організаційної структури використовується ряд **правил**, які зберігаються в репозиторії.

1. Правила цілісності моделі організації.

Правила цілісності використовуються для перевірки моделі підприємства на несуперечність і повноту бізнес-функцій. Після того, як модель підприємства створена, можна перевірити її погодженість, оцінюючи всі правила цілісності.

Оскільки ці правила можуть бути визначені всередині моделі функцій, то можливим стає динамічне конфігурування бізнес-процесу.

Приклад.

Якщо є варіант бізнес-функції "Пряме постачання", то в моделі повинні бути представлені бізнес-функції "Обробка замовлення на придбання" і "Обробка замовлення на збут".

2. Правила перетворення моделей функцій у моделі процесів.

Модель бізнес-функції може бути автоматично перетворена в модель бізнес-процесу за допомогою правил перетворення. Ці правила вказують на відповідність бізнес-функції бізнес-процесу.

Приклад.

Якщо був визначений варіант бізнес-функції "Обробка замовлення на придбання з контрактами", необхідно вибрати бізнес-процес "Обробка контрактів".

3. Правила конфігурації (встановлення параметрів).

Правила конфігурації використовуються для присвоєння значення параметра залежно від його наявності в бізнес-функціях, бізнес-процесах чи їхніх комбінаціях.

Приклад.

Якщо був визначений варіант бізнес-функції "Обробка замовлення на покупку в електронному обміні даними", то значення параметра "Електронний обмін даними" настраюється на "так".

4. Правила встановлення статичних умов.

Контроль є одним із компонентів бізнес-процесу і використовується для моделювання варіантів бізнес-процесу. Правила встановлення статичних умов використовуються для вибору варіанта бізнес-процесу залежно від виконання умови. Якщо значення логічної умови "Неправда", то даний варіант (гілка) бізнес-процесу не виконується, і така неактивна гілка бізнес-процесу подається в затемненому вигляді.

Приклад.

Якщо функція "Оформлення акредитива" не використовується у фазі впровадження, то гілка "Оформлення акредитива" забороняється і зображується в затемненому вигляді

13.3.6. Моделі об'єктів (даних)

У модельно-орієнтованій технології проектування ІС інтегрування різних бізнес-процесів (прикладних програм) здійснюється на основі бізнес-об'єктів. Відповідно до визначення комітету Business Object Task Force OMG **бізнес-об'єкти** – це компоненти рівня предметної області, що використовуються в різних додатках у довільних комбінаціях і не залежать від них. При цьому додаток забезпечує середовище для функціонування бізнес-об'єктів. З одного боку, бізнес-об'єкти – це об'єктності в нотації UML, наприклад замовлення, рахунки, матеріали, постачальники і т. д. З іншого боку, на відміну від звичайних об'єктностей бізнес-об'єкти є **самодостатніми**, тобто мають стандартний інтерфейс, написаний мовою опису інтерфейсів IDL (Interface Definition Language), за допомогою якого бізнес-об'єкти можуть взаємодіяти один з одним через об'єктну шину – брокер об'єктних запитів (Object Request Broker). Таким чином, бізнес-об'єкти мають більш складну внутрішню структуру порівняно з простими об'єктами.

13.3.7. Модель організаційної структури

Модель організаційної структури підприємства становить традиційну ієрархічну структуру підпорядкування підрозділів і персоналу (організаційних одиниць). Призначення моделювання організаційної структури стосовно до інформаційної системи полягає в розподілі функцій, що автоматизуються, між працівниками підрозділів і визначенні повноважень доступу до інформаційної системи.

Модель організаційної структури має таке призначення:

одержати уявлення про поточну і, в міру можливості, майбутню (цільову) організаційну структуру і задокументувати їх;

одержати уявлення про зміни в структурі підрозділів підприємства залежно від розташування їх у загальній системі і задокументувати їх;

уможливити створення інтерфейсу користувача корпоративної системи в розрізі працівників або виконуваних функцій, що полегшується накладенням організаційної моделі на модель процесів.

Модель організаційної структури має такі характеристики:

модель організаційної структури складається з ряду організаційних компонентів. Ці організаційні компоненти можна пов'язати з ім'ям, типом і атрибутами;

між організаційними компонентами можуть бути встановлені функціональні відносини. З такими функціональними відносинами можна пов'язати текст;

модель організаційної структури може складатися з однієї чи більше організаційних схем. Дані схеми можуть співвідноситися одна з одною за допомогою визначення компонентів субдіаграм;

користувачі та функції, які ними виконуються, можуть співвідноситися з організаційними компонентами;

моделі організаційної структури можуть визначатися за фазами оптимізації накладенням бізнес-процесів на організацію для того, щоб створити наочне уявлення про проект.

Структура персоналу може бути описана на наступних рівнях:

на абстрактному рівні за допомогою опису ролей;

на конкретному рівні – зіставленням працівників і організаційних одиниць.

Роль виступає у вигляді функції, що може бути реалізована тільки тими працівниками, за якими ця функція закріплена. Ролі і, якщо необхідно, обов'язки можуть бути пов'язані з роботами, бізнес-процесами, організаційними компонентами і бізнес-функціями. За допомогою ролей визначається, які працівники (групи працівників) уповноважені на виконання визначеної роботи (чи мають відповідні обов'язки).

Розрізняють два типи ролей:

за організаційними одиницями – це ролі, які є функціями працівників, що пов'язані з організаційними одиницями в специфічній для замовника організаційній діаграмі, наприклад: бухгалтер; плановик; фахівець із купівель; фахівець із продажів; менеджер; комірник;

за бізнес-функціями – це ролі, які є функціями працівників, що виконуються в рамках визначених бізнес-функцій, наприклад: а) бізнес-функція: продаж, ролі: фахівець із продажів, секретар; б) бізнес-функція: забезпечення матеріалами, ролі: менеджер, комірник.

Обов'язок становить певне завдання, за допомогою якого специфікується роль (тобто функція, виконувана групою працівників), наприклад: "Консультавання"; "Інформування"; "Ухвалення рішення"; "Виконання";

"Контроль виконання".

З роботою може бути пов'язано не більше шести ролей. Для кожної ролі можна задати не більше шести обов'язків.

Ролі й обов'язки успадковуються на нижніх рівнях бізнес-процесів. Якщо для певного бізнес-процесу одна роль використовується для всіх обов'язків у всіх роботах, то досить пов'язати цю роль і відповідні обов'язки з даним процесом. Спадкування ролей і обов'язків не застосовується до організаційних компонентів і бізнес-функцій.

За допомогою інструмента BEW у моделі бізнес-процесу явно вказується закріплення автоматизованих функцій за організаційними одиницями: підрозділами або посадами (рис. 63).

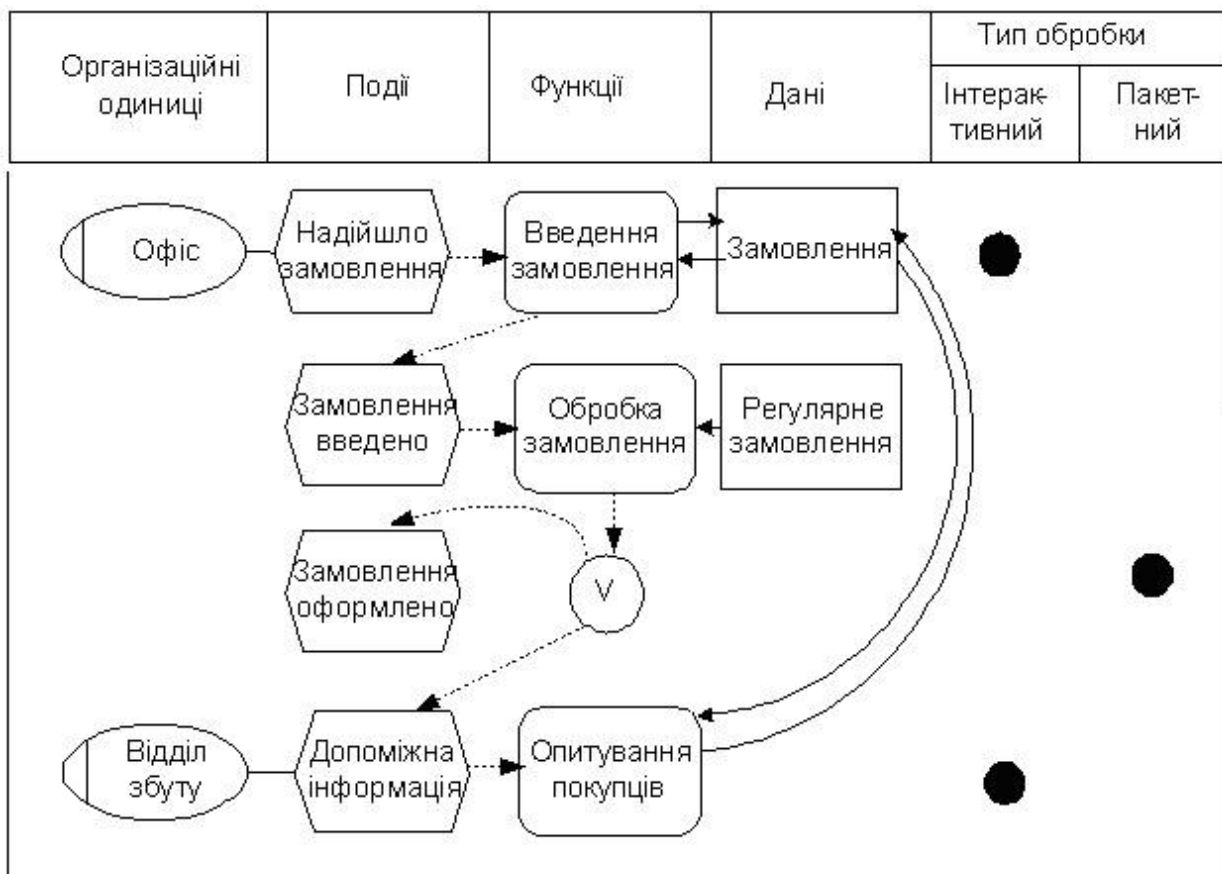


Рис. 63. Явне закріплення ролей у моделі бізнес-процесу (система R/3)

За допомогою інструменту Enterprise Modeler зв'язок моделі бізнес-процесу і моделі організаційної структури задається через покажчики ролі, що можуть виконуватися різними організаційними одиницями. Через покажчики ролі співробітників устанавлюється зв'язок організаційної структури з бізнес-процесами (рис. 64). Покажчик ролі визначає тип працівника, що може виконувати ту або іншу роботу (економіст, бухгалтер, менеджер і т. д.). Для кожної ролі визначаються

повноваження у виконанні функцій, права доступу до інформації, посадові інструкції.

Під час призначення роботи конкретному працівнику завжди здійснюється перевірка ролі, яку він може виконувати. Якщо при цьому тип конкретного працівника не відповідає ролі, то останній не одержує доступу до виконання.

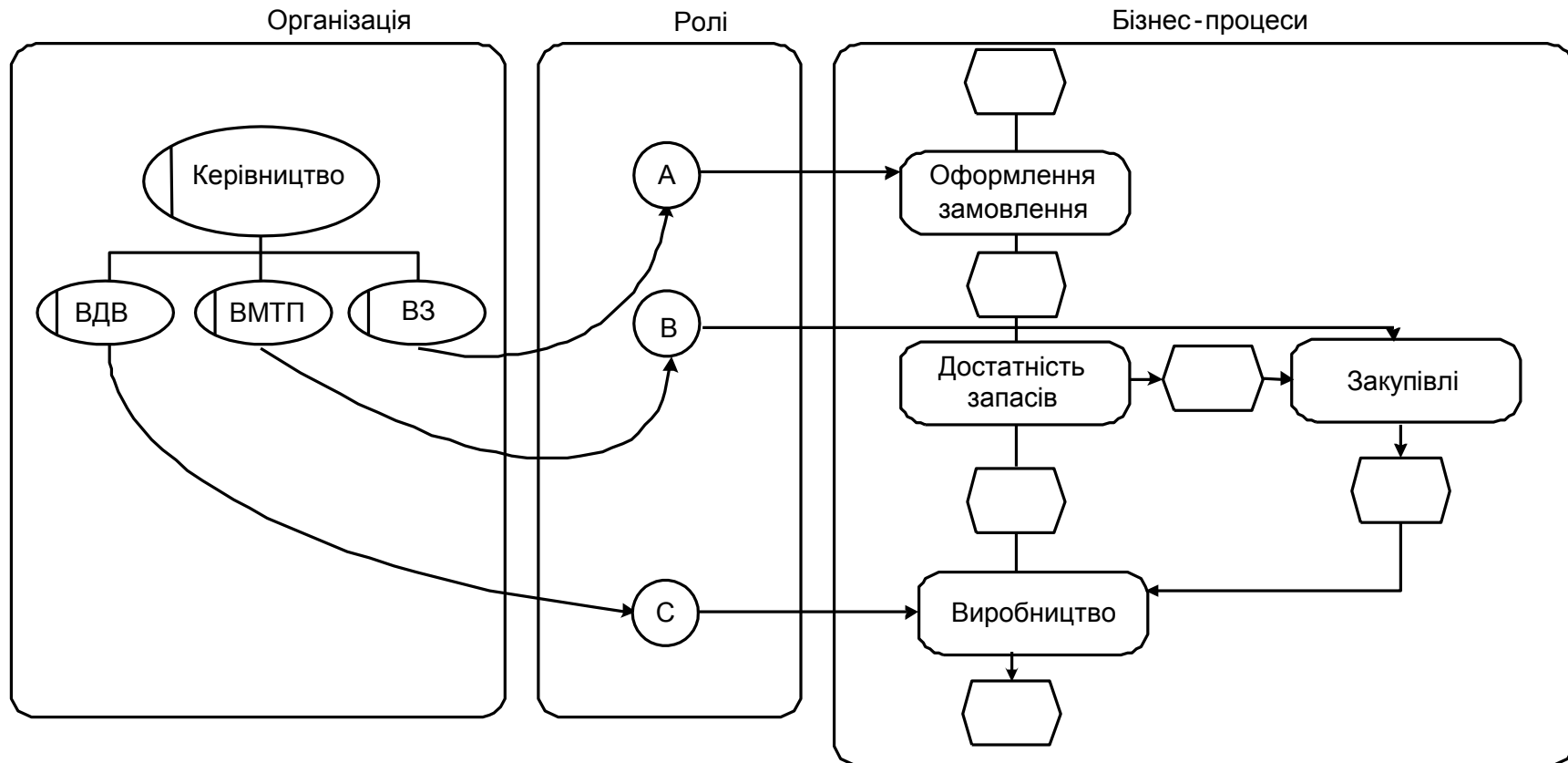


Рис. 64. Установлення зв'язку моделі організаційної структури і моделі бізнес-процесу за допомогою ролей (система BAAN)

Умовні позначення:

ВДВ – виробничо-диспетчерський відділ;

ВМТП – відділ матеріально-технічного постачання;

ВЗ – відділ збуту.

13.3.8. Технологія модельно-орієнтованого проектування

Складність типової інформаційної системи обумовлює характерні риси типового модельно-орієнтованого проектування. До них можна віднести наступні особливості:

прив'язка типової ІС до умов конкретної організації здійснюється спільними зусиллями фірми-виробника програмного продукту і проектною групою підприємства;

консультанти з боку фірми-виробника програмного забезпечення беруть участь на всіх етапах упровадження системи і особливо на етапі аналізу вимог;

зростає роль керівника підприємства в організації і контролі за створенням ІС.

У методологіях модельно-орієнтованого проектування ІС велика увага приділяється регламентації послідовності операцій на основі застосування програмних засобів планування, що дозволяють прискорити процес впровадження типової ІС.

Технологія модельно-орієнтованого проектування включає чотири основні етапи:

вибір типового проекту;

розробку проектною моделі організації;

реалізацію проекту;

введення в експлуатацію і підтримку функціонування.

На всіх етапах використовується інструментарій моделювання предметної області організації. Слід розглянути далі докладно етапи модельно-орієнтованого проектування.

1. Вибір типової ІС – аналіз вимог.

Впровадження типової ІС починається з аналізу вимог конкретної організації до ІС. Зокрема, на основі результатів передпроектного обстеження формується попередня модель підприємства, яка акумулює вимоги до функціональності ІС (до множини функцій, що автоматизуються) і на основі якої здійснюється вибір найбільш адекватного програмного комплексу. Дана робота може бути виконана в рамках проведення попереднього реінжинірингу бізнес-процесів.

Можливий також інший варіант аналізу вимог, що визначаються наявною організацією бізнес-процесів. У другому випадку реально існуюча модель підприємства буде адаптуватися (еволюціонувати) на етапі експлуатації

інформаційної системи з метою оптимізації функціонування організації. На думку фахівців, такий еволюційний інжиніринг забезпечує більш швидке впровадження корпоративної ІС, скорочення капітальних витрат і підвищення ефективності експлуатації ІС.

Організація, що впроваджує типову інформаційну систему, прагне вибрати найкращий програмний продукт, що відповідає конкретним умовам. Тому на етапі вибору типового проекту в конкурсі може брати участь кілька фірм-конкурентів, програмні продукти не тільки оцінюються за методикою, розглянутою в п. 13.2.2, але й з позиції реалізації запропонованої моделі підприємства. У результаті аналізу вимог вибирається конкретна типова ІС.

2. Розробка проектної моделі організації.

Для модельно-орієнтованої технології проектування ІС характерна прив'язка моделі організації до функціональності типової ІС, на основі якої надалі автоматично виконується конфігурація інформаційної системи. Тому на вході задається попередня модель організації, референтна модель функціональності типової ІС і бізнес-правила відображення моделі, а на виході формується проектна модель підприємства, у якій визначаються компоненти типової ІС, компоненти інших програмних продуктів і компоненти, що повинні бути спеціально розроблені за допомогою інструментальних засобів, наприклад АВАР/4 (SAP), чи набори інструментів Tools (BAAN).

У процесі розробки проектної моделі підприємства виконуються наступні роботи:

- інсталяція програмного продукту, що реалізує типову ІС;

- проведення навчання проектної команди;

- прив'язка моделі підприємства до компонентів типової інформаційної системи;

- визначення вимог до доробки програмного забезпечення;

- проектування зовнішніх інтерфейсів системи.

3. Реалізація типового проекту ІС, введення в експлуатацію і підтримка функціонування. Реалізація типового проекту ІС зводиться до конфігурації ІС і генерації інтерфейсів користувачів, тобто одержання готових для експлуатації програм функцій обробки даних та інтерфейсів.

Реалізація типового проекту передбачає встановлення наступних параметрів:

- глобальних параметрів;

- структури компанії;

- структури основних даних;

функцій і процесів;
інтерфейсів;
системи звітів;
системи архівування;
системи авторизації доступу.

Під час завершення стадії реалізації здійснюється комплексне тестування усіх компонентів ІС.

4. Введення в експлуатацію і підтримка функціонування.

Введення в експлуатацію типового проекту здійснюється поетапно відповідно до визначеного плану. Перед початком експлуатації повинні бути виконані наступні роботи:

створення документації кінцевих користувачів і їх навчання;
встановлення програмно-технічного середовища експлуатації ІС;
наповнення інформацією нових БД або підключення і конвертація наявних БД.

У процесі експлуатації здійснюється системна підтримка для усунення виниклих зауважень. Особливу увагу приділяють розвитку проекту ІС. Для цього система повинна накопичувати статистику про характер функціонування ІС, на основі якої провадиться технологічне налагодження ефективності експлуатації ІС. Важливо також здійснювати аналіз ефективності організації за допомогою інформаційної системи бізнес-процесів на основі контролінгу економічних показників, який призводить до безперервного вдосконалення проектної моделі підприємства і до адаптації ІС до необхідних змін.

Контрольні запитання

1. Що таке типове проектне рішення?
2. Які виділяють методи типового проектування?
3. Що є типовим елементом при елементному методі проектування?
4. Охарактеризуйте склад ТПР для завдання.
5. Назвіть переваги, недоліки і область застосування елементного методу проектування.
6. Назвіть переваги і недоліки підсистемного методу проектування.
7. Наведіть приклади функціональних ППП.
8. Які вимоги висуваються до типових проектів?
9. Що є типовим елементом за об'єктного методу проектування?
10. Порівняйте існуючі підходи до реалізації об'єктного методу проектування. Наведіть приклади типових систем.

11. Наведіть структурну схему ППП, що параметрично настроюється. Охарактеризуйте основні його складові. Як настроюється і функціонує такий ППП?
12. Охарактеризуйте основні етапи параметричного настроювання ППП.
13. Перелічіть основні групи критеріїв вибору ППП. Наведіть приклади окремих критеріїв.
14. Опишіть методику оцінювання ППП.
15. Що таке репозиторій типової ІС, яке його призначення?
16. Що таке базова модель, які її складові?
17. Що таке типова модель, модель організації?
18. Наведіть схему конфігурації ІС за модельної технології типового проектування.
19. Які підходи використовуються в процесі моделювання організації?
20. Що таке модель функцій, які її складові?
21. На прикладі моделі функцій наведіть її характеристики.
22. Наведіть класифікацію бізнес-функцій та охарактеризуйте кожний з її рівнів.
23. Що таке модель процесів? Яке її призначення, основні характеристики?
24. Які існують види бізнес-процесів?
25. Що таке модель правил? Наведіть приклади правил.
26. Що таке бізнес-об'єкти? Порівняйте їх з об'єктами-сутностями.
27. Що таке модель організаційної структури? Яке її призначення?
28. Яке призначення і характеристики моделі організаційної структури?
29. Дайте визначення понять "роль", "обов'язок". Які виділяють типи ролей? Наведіть приклади ролей і обов'язків.
30. Яким чином відбувається успадкування ролей і обов'язків в моделях процесів, функцій, організаційної структури?
31. На яких рівнях може бути описана структура персоналу? Наведіть приклади закріплення ролей.
32. Охарактеризуйте основні етапи технології модельно-орієнтованого проектування.

14. Управління проектуванням ІС

14.1. Особливості проектної діяльності

Проектування як вид діяльності зазвичай називають проектом. Термін "проект" походить від латинського слова *projectus* (буквальний переклад – *кинутий вперед*) і використовується для зазначення технічних документів (креслень, розрахунків, макетів новостворюваних будівель, споруд, машин,

приладів і т. д.), планів, задумів і т. д. Під **проектом** зазвичай мається на увазі результат певного досить складного виду діяльності, управління яким є також досить складним.

У міжнародних стандартах даються наступні визначення проекту:

діяльність, спрямована на розробку нового продукту або поліпшення наявного (ISO/IEC FDIS 18019:2003);

часове зусилля, використовуване для того, щоб створити унікальний продукт або послугу з певною датою початку і закінчення дії, що відрізняється від дій, які продовжуються, повторних і що вимагають прогресивного вдосконалення характеристик (PMI).

Суттєвими характеристиками проекту як виду діяльності є наступні.

1. Мета проекту. Кінцевий результат, вихід, продукція, яка визначається в термінах витрат, якості та часу реалізації.

2. Складність. Для досягнення цілей проекту необхідно вирішити безліч взаємопов'язаних завдань, відносини між якими можуть бути досить складними, особливо, якщо в проекті багато завдань.

3. Життєвий цикл. У міру реалізації проекту змінюється потреба в тих або інших ресурсах. Ця зміна відбувається в певній передбаченій послідовності.

4. Обмеженість у часі. Проект має початок і кінець. Для його реалізації необхідна тимчасова концентрація ресурсів. Після його завершення, ресурси використовуються на інші цілі.

5. Обмеженість ресурсів. Ресурси, що виділяються на виконання проекту – фінансові, трудові та матеріальні, мають обмеження, передбачені бюджетом проекту.

6. Невизначеність. Можливість досягнення мети в зазначені терміни з виділеними ресурсами заздалегідь не гарантована.

7. Передбаченість. У міру реалізації проекту змінюється потреба в тих або інших ресурсах. Ця зміна відбувається в певній передбаченій послідовності, яка визначається життєвим циклом проекту.

8. Унікальність. Проект – це разові дії, які не повторюються. Проекти, навіть ті, що "повторюються", наприклад, щодо впровадження ще однієї ІС за тією ж проектною документацією, значно відрізняються один від одного ресурсами, що використовуються, і середовищем реалізації.

Проекти можуть сильно відрізнятися за сферою застосування, складом, предметною областю, масштабами, тривалістю, складом учасників, ступенем складності, значущістю результатів і под. Тому проекти можуть бути класифіковані за різними ознаками. Виділимо основні з них.

Клас проекту визначається за складом і структурою проекту. Зазвичай розрізняють:

монопроект (окремий проект, який може бути будь-якого типу, виду і масштабу);

мультипроект (комплексний проект, що складається з ряду монопроектів і вимагає застосування багатопроєктного управління).

Тип проекту визначається за основними сферами діяльності, в яких здійснюється проект. Можна виділити п'ять основних типів проекту:

- технічний;
- організаційний;
- економічний;
- соціальний;
- змішаний.

Розробка інформаційних систем відноситься, переважно, до технічних проектів, які мають наступні особливості:

головна мета проекту чітко визначена, але окремі цілі повинні уточнюватися в міру досягнення окремих результатів;

термін завершення і тривалість проекту визначені заздалегідь, бажане їх точне дотримання, однак вони також можуть коректуватися залежно від отриманих проміжних результатів і загального прогресу проекту.

Масштаб проекту визначається розмірами бюджету і кількістю учасників:

- дрібні проекти;
- малі проекти;
- середні проекти;
- великі проекти.

Можна також розглядати масштаби проектів в конкретнішій формі – галузеві, корпоративні, відомчі проекти, проекти одного підприємства.

Загальний вигляд проекту як виду діяльності наведений на рис. 65.

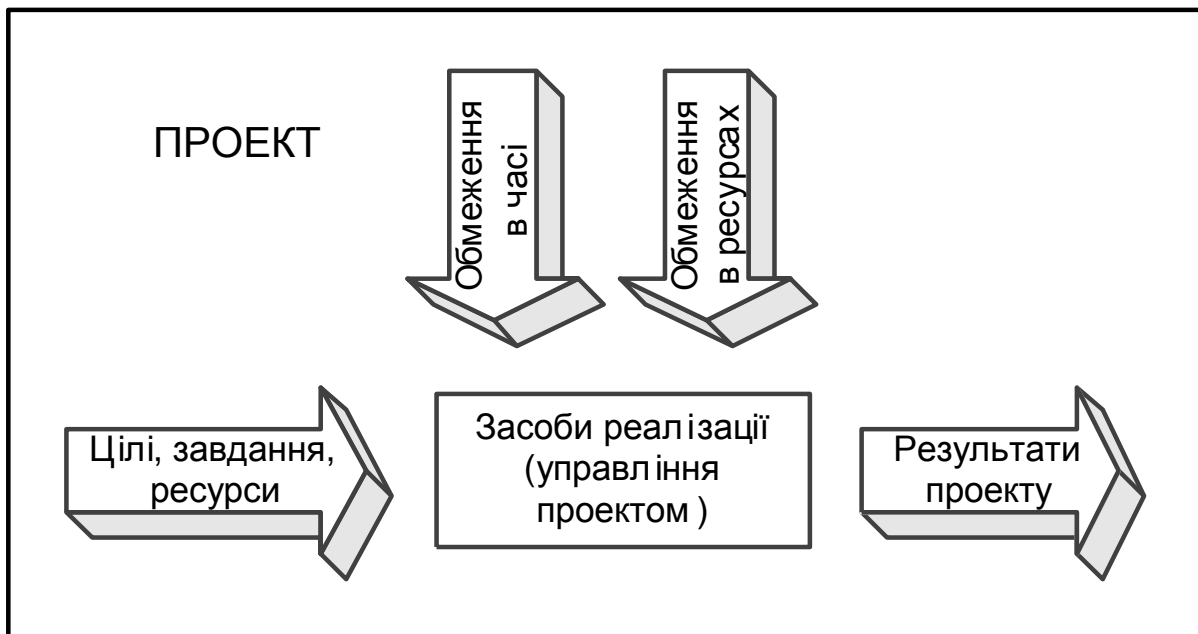


Рис. 65. **Загальна схема проекту**

Процес проектування ІС включає велику кількість взаємозалежних між собою різноманітних елементів і передбачає побудову відповідної структури організації цього процесу. Як об'єкт розробки проекту ІС можуть виступати:

вся ІС для підприємства замовника;

окрема підсистема;

сукупність підсистем;

окремі роботи, наприклад встановлення обчислювальної мережі, проведення консалтингових робіт з оцінки ефективності інформаційної системи і т. д.

Отже, проект – це досить складний вид діяльності, яким складно управляти через його унікальність і обмеженість ресурсів і часу. Ця обставина вносить до проекту елемент невизначеності, а правильно організоване управління проектом робить результати передбачуваними. Однак передбачуваний проект ще не означає успішний. Передбачуваний проект означає вчасно завершений, тобто успішний проект або вчасно припинений, тобто проект неуспіху.

14.2. Основні складові управління проектами

Існує декілька визначень управління проектами (Project Management, PM): набір перевірених принципів, методів і методик, використовуваних для ефективного планування, складання графіка, управління і відстежування результатів роботи (PMI®);

спеціалізація загального менеджменту, що визначає застосування стандартних керівних навичок планування, організації, комплектування персоналом, просування, а також управління і контролю для досягнення певної мети проекту [12];

наука і мистецтво управління і координації трудових та матеріальних ресурсів упродовж життєвого циклу проекту шляхом застосування сучасних методів і техніки управління для досягнення визначених у проекті результатів за складом і обсягом робіт, вартістю, часом, якістю і задоволенням учасників проекту (РМВОК).

Таким чином, під **управлінням проектом** мається на увазі діяльність, спрямована на реалізацію проекту з максимально можливою ефективністю за заданих обмежень у часі, у коштах і матеріальних ресурсах, а також за якістю кінцевих результатів проекту (документованих, наприклад, у технічному завданні).

Управління проектом як процес характеризується наступними компонентами:

- метою управління;
- обмеженнями;
- об'єктом управління (проектом);
- суб'єктом управління (командою).

Обмеженнями можуть виступати терміни проектування, необхідні ресурси – фінансові, матеріальні, трудові тощо.

Обмеження – це ключовий компонент управління проектами. Основним завданням виконання проекту є забезпечення виконання робіт у строк, в рамках виділених засобів, відповідно до технічного завдання. Саме ці три моменти – час, бюджет і якість робіт – знаходяться під постійною увагою керівника проекту. Їх також можна назвати основними обмеженнями, що накладаються на проект.

Ці три основні обмеження (терміни, витрати і якість результату) взаємопов'язані. Для ілюстрації взаємозв'язку використовують трикутник обмежень, у якому якість, час і витрати інтерпретуються площами внутрішніх трикутників (рис. 66). У цьому трикутнику центр і верхня вершина фіксовані, а нижні вершини можуть переміщуватися. Трикутник ілюструє, що будь-яке скорочення фінансів або часу веде до скорочення якості, а збільшення якості може бути досягнуте за рахунок збільшення фінансування або термінів.

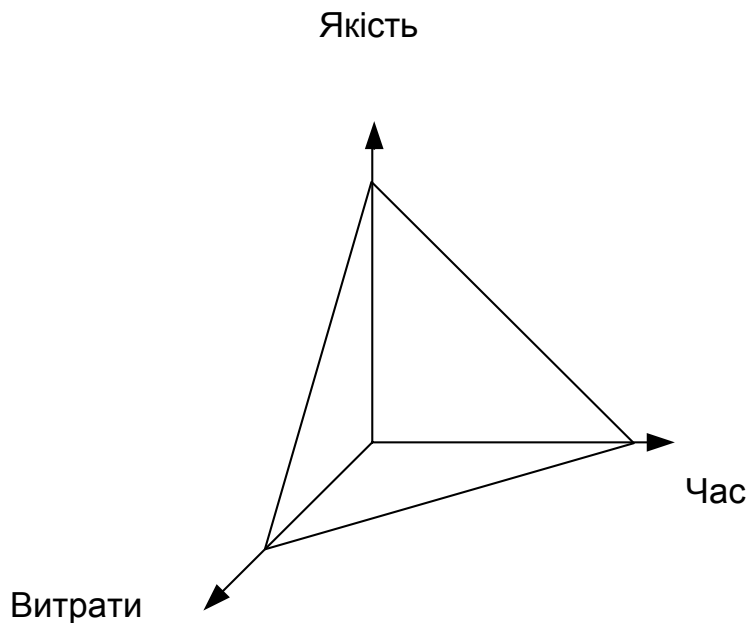


Рис. 66. Трикутник обмежень в управлінні проектами

Об'єктом управління є процес проектування ІС як діяльність колективу розробників системи, а також стан використовуваних ресурсів. Процес проектування ІС має специфічні особливості, що, у свою чергу, визначають специфіку організації та управління проектуванням.

1. Процес проектування ІС за своїм характером є творчим. Тому за відсутності досить повного формалізованого переліку операцій проектування і станів проекту в процесі його розробки управління проектуванням має ситуаційний характер.

2. Користувач на етапі розробки системи може змінювати вимоги до якості системи, термінів і витрат проектування. У зв'язку з відсутністю загальноприйнятих надійних способів оцінки якістю проектних рішень ускладнений його контроль.

3. Прагнення розробників до індивідуального характеру праці приводить до невисокого ступеня організації контролю і координації діяльності окремих розробників проекту.

Стан проекту від ідеї до завершення характеризується зміною ряду показників, які визначають його суть і на основі яких встановлюється успішність проекту. Ця сукупність показників ("елементів проекту"), по суті, і є об'єктами управління. Таких об'єктів управління в кожному проекті може бути досить багато. Однак для всіх проектів можна виділити найбільш суттєві об'єкти управління (рис. 67). Далі охарактеризуємо кожен із цих об'єктів.

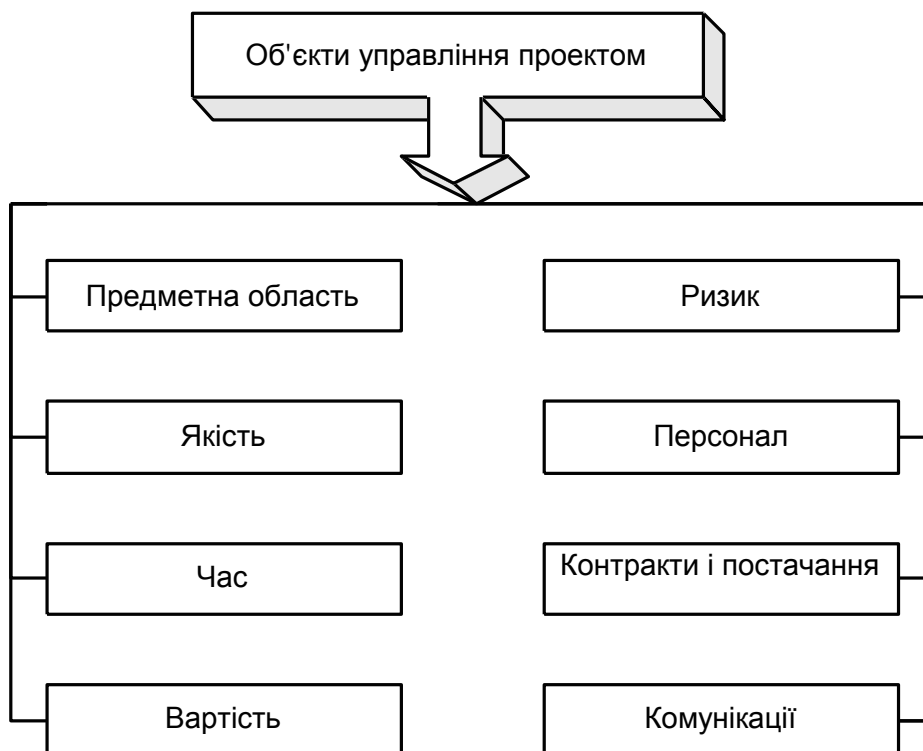


Рис. 67. **Об'єкти управління проектом**

Предметну область проекту визначають цілі, результати і склад робіт.

У процесі життя проекту всі складові предметної області зазнають змін: цілі, результати і склад робіт можуть змінюватися або уточнюватися як у процесі розробки проекту, так і в міру досягнення проміжних результатів;

обсяги робіт можуть уточнюватися в процесі розробки проекту, а в процесі його життєвого циклу вони змінюються від нуля до 100%.

Управління предметною областю полягає в управлінні змінами, що відбуваються в ній, упродовж життєвого циклу проекту і здійснюється через процеси: ініціації робіт, планування й визначення предметної області, уточнення і підтвердження.

Управління якістю проекту проходить через весь життєвий цикл проекту, охоплює всі його сторони й елементи і включає:

проектні, організаційні, управлінські рішення та ін.;

якість результатів проекту ІС.

Управління якістю реалізується через встановлення вимог і стандартів до якості результатів проекту, планування якості, забезпечення виконання вимог до якості в процесі реалізації проекту через систему тестування, верифікації і т. д.

Управління часом включає:

визначення характеру робіт і їх тривалості, термінів початку і завершення проекту, його фаз, контрольних подій (віх) і кожної з виконуваних робіт;
мінімізацію часових характеристик;
розумне використання резервів часу;
контроль проекту за тимчасовими характеристиками;
прогнозування термінів завершення робіт, етапів і проекту в цілому;
ухвалення рішень щодо ліквідації небажаних тимчасових відхилень.
Управління часом здійснюється шляхом аналізу термінів виконання проекту і його частин, календарного планування робіт, їх актуалізації і коректування.

Управління вартістю включає планування ресурсів, попередню оцінку витрат, пов'язаних із проектом, визначення кошторису витрат, грошових потоків, прогнозування доходів і прибутків, контроль витрачання і надходження грошових коштів і ухвалення рішень у разі перевищення витрат та інших відхилень від фінансових планів. Головним завданням управління вартістю є дотримання бюджетних рамок проекту і отримання передбаченого прибутку від його здійснення.

Ризик у контексті проекту розглядається як вплив на проект і його елементи непередбачених подій, які можуть завдати певного збитку і перешкоджати досягненню цілей проекту. Ризик проекту характеризується трьома чинниками: подіями, що негативно впливають на проект; вірогідністю виникнення таких подій; оцінкою шкоди, нанесеної проекту такими подіями. **Управління ризиком** – це мистецтво і формальні методи прогнозування, аналізу, оцінки, попередження виникнення ризикових подій; вживання заходів щодо зниження ступеня ризику впродовж життя проекту і розподілу можливої шкоди від ризиків між учасниками проекту.

Управління персоналом у проекті включає:

визначення потреби, чисельного і кваліфікаційного складу персоналу на всі періоди здійснення проекту;
пошук і відбір кандидатур, оформлення прийому на роботу і звільнення; планування і розподіл працівників за робочими місцями;
організацію навчання й підвищення кваліфікації;
встановлення відповідальності; створення умов і робочої атмосфери колективної роботи;
запобігання і вирішення конфліктів;
встановлення оплати праці та ін.

Управління персоналом повинне бути спрямоване на оптимальне використання трудових ресурсів для досягнення цілей проекту.

Управління контрактами і постачаннями в проекті включає:
вибір стратегії контрактної діяльності;
визначення складу, номенклатури і термінів роботи суб'єктів, що залучаються за контрактом;
підготовку контрактних пропозицій;
вибір контрагентів і постачальників шляхом торгів, конкурсів, тендерів та ін.;
підготовку документації для укладання контрактів;
контроль за виконанням контрактів;
закриття і розрахунок за завершеними контрактами.

Управління комунікаціями включає розробку, організацію і контроль процесу інформаційного обміну за допомогою різноманітних засобів для задоволення потреб учасників проекту. Управління проектом у цілому залежить від успішної організації взаємодії всіх учасників проекту і забезпечення їх потреби в інформації для здійснення проекту. Масштаби, складність проекту, його середовище, кількість учасників і зацікавлених сторін породжують велику кількість взаємозв'язків і потоків інформації.

Суб'єкт управління виділяється відповідно до поділу праці в групі фахівців у процесі проектування ІС. Суб'єктами управління проектними роботами залежно від рівня управління в цьому випадку можуть бути:

- керівники проектної організації;
- керівники забезпечувальних підрозділів (наприклад, планово-виробничого відділу і т. д.);
- керівники функціональних підрозділів;
- керівники проектів (головні конструктори);
- керівники проектних груп (відповідальні виконавці).

На кожному рівні управління проектними роботами існує чітке уявлення про процес проектування, окремі цілі й завдання управління процесом проектування ІС, що визначається колом посадових обов'язків, характером виконуваних функцій суб'єктів управління кожним рівнем, набором використовуваних методів і засобів управління.

14.3. Процеси управління проектами

Управління проектуванням ІС у функціональному аспекті розглядається як сукупність взаємозалежних процесів. Під **процесами управління** маються на увазі дії і процедури, пов'язані з вирішенням конкретних завдань або реалізацією функцій управління, до яких належать процеси:

ініціації, пов'язані з ухваленням рішення про початок виконання проекту, якого-небудь чергового етапу чи його фази;

планування – сукупність процедур, пов'язаних із визначенням цілей і критеріїв успіху проекту і розробкою робочих схем їх досягнення;

виконання, призначені для координації людей та інших ресурсів для виконання плану;

аналізу, що дають можливість визначити відповідність плану і виконання проекту поставленим цілям і критеріям успіху і ухвалити рішення про необхідність застосування коригувальних впливів;

оперативного управління (регулювання) – сукупність процедур, призначених для визначення необхідних коригувальних впливів, їх узгодження і застосування;

завершення – процеси формалізації виконання проекту і складання звітності.

Процеси управління проектами накладаються один на одного і відбуваються з різними інтенсивностями на всіх фазах проекту (рис. 68).

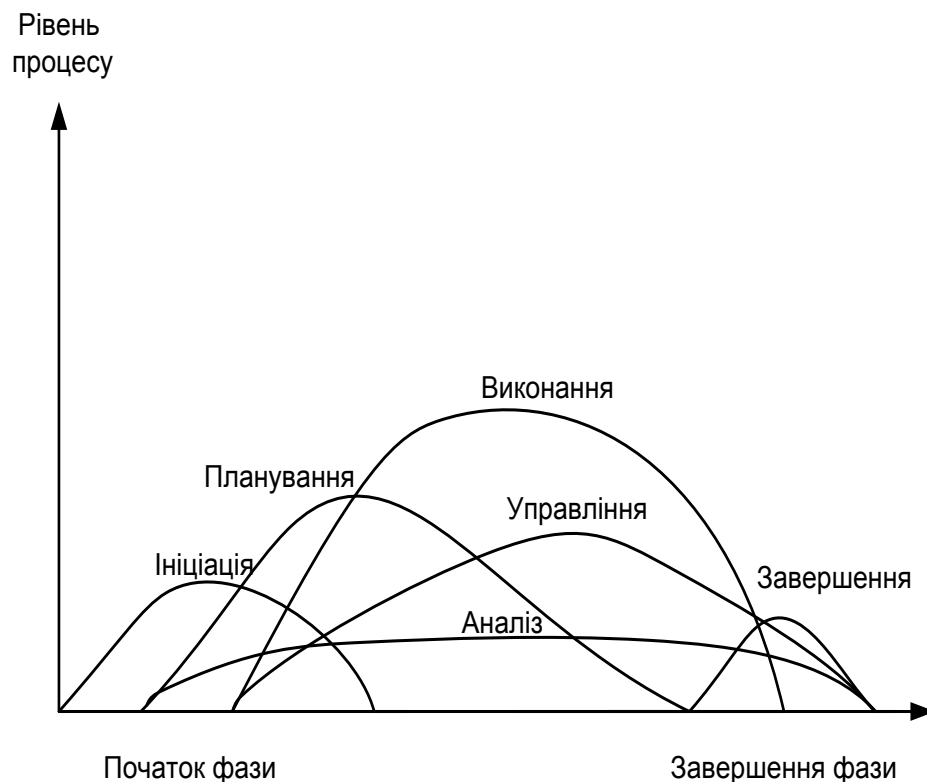


Рис. 68. Накладання груп процесів у фазі

Крім того, процеси управління проектами пов'язані між собою своїми результатами: результат виконання одного стає вхідною інформацією для іншого (рис. 69). Нарешті, існують взаємозв'язки груп процесів різних фаз (етапів) проекту. Наприклад, закриття однієї фази може бути входом для ініціації наступної фази (приклад: завершення фази проектування вимагає схвалення замовником проектної документації, яка необхідна для початку реалізації). У реальному проекті фази можуть не тільки передувати одна одній, але й накладатися. Усередині кожної групи процеси управління проектами пов'язані один з одним через свої:

входи – документи чи документовані показники, згідно з якими процес виконується;

виходи – документи чи документовані показники, що є результатом процесу;

методи й засоби – механізми, за якими вхід перетвориться у вихід.

Слід розглянути докладніше групи процесів управління проектами.

1. Процеси ініціації.

Ініціація включає єдиний підпроцес – авторизацію, тобто дозвіл почати наступну фазу проекту.

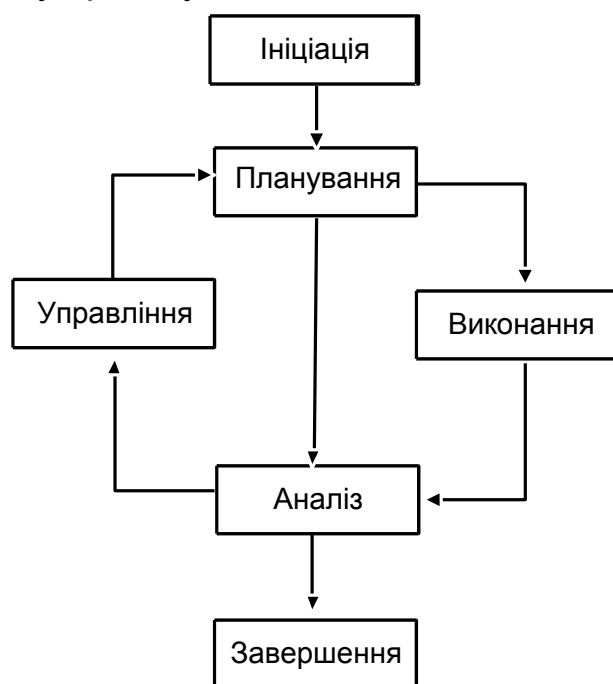


Рис. 69. Взаємозв'язок груп процесів управління проектами у фазі

2. Процеси планування.

Планування має велике значення для проекту і включає порівняно багато процесів. Деякі з процесів планування мають чіткі логічні й інформаційні взаємозв'язки і виконуються в одному порядку практично у всіх проектах. Так, наприклад, спочатку варто визначити, з яких робіт складається проект, а потім розраховувати терміни виконання і вартість проекту. Ці основні процеси виконуються по кілька разів протягом кожної фази проекту.

До основних процесів планування проектних робіт відносяться:

планування цілей – розробка постановки завдання (проектне обґрунтування, основні етапи й цілі проекту);

декомпозиція цілей – поділ етапів проекту на більш дрібні і більш керовані компоненти для забезпечення більш дієвого контролю;

визначення складу операцій (робіт) проекту – визначення переліку операцій, з яких складається виконання різних етапів проекту;

визначення взаємозв'язків операцій – складання і документування технологічних взаємозв'язків між операціями;

оцінка тривалості чи обсягу робіт – оцінка кількості робочих часових інтервалів або обсягів робіт, необхідних для завершення окремих операцій;

визначення ресурсів (людей, устаткування, матеріалів) проекту – визначення загальної кількості ресурсів усіх видів, які можуть бути використані на роботах проекту (ресурсів організації), і їх характеристик;

призначення ресурсів – визначення ресурсів, необхідних для виконання окремих операцій проекту;

оцінка вартості – визначення складових вартості операцій проекту й оцінка цих складових для кожної операції, ресурсу і призначення;

складання розкладу виконання робіт – визначення послідовності виконання робіт проекту, тривалості операцій і розподілу в часі потреб у ресурсах і витрат з урахуванням накладених обмежень і взаємозв'язків;

оцінка бюджету – визначення оцінок вартості до окремих компонентів проекту (етапів, фаз, термінів);

розробка плану виконання проекту – інтеграція результатів інших підпроцесів для складання повного документа;

визначення критеріїв успіху – розробка критеріїв оцінки виконання проекту.

Крім перерахованих основних процесів планування є ряд допоміжних процесів, необхідність у використанні яких значно залежить від природи конкретного проекту. Такі процеси включають:

планування якості – визначення того, які стандарти якістю слід використовувати в проекті і як цих стандартів досягти;

планування організації – визначення, документування і призначення ролей, відповідальності та взаємин підпорядкування в організації;

призначення персоналу – призначення людських ресурсів на виконання робіт проекту;

планування взаємодії – визначення потоків інформації та способів взаємодії, необхідних для учасників проекту;

ідентифікація ризику – визначення і документування подій ризику, що можуть вплинути на проект;

оцінка ризику – оцінка ймовірностей настання подій ризику, її характеристик і впливу на проект;

розробка методів реагування – визначення необхідних дій для попередження ризиків і реакції на загрозливі події;

планування постачань – визначення того, що, як і коли повинно постачатися;

підготовка умов – вироблення вимог до постачань і визначення потенційних постачальників.

Взаємозв'язки між допоміжними підпроцесами, як і сама їхня наявність, значною мірою залежать від природи проекту.

3. Процеси виконання і контролю.

Під **виконанням** маються на увазі процеси реалізації складеного плану. Виконання проекту повинно регулярно вимірюватися й аналізуватися для того, щоб виявити відхилення від наміченого плану й оцінити їхній вплив на проект. Регулярний вимір параметрів проекту й ідентифікація виниклих відхилень відносяться до процесів виконання й іменуються **контролем виконання**. Контроль виконання варто проводити за всіма параметрами, що входять у план проекту.

Як і в плануванні, процеси виконання можна підрозділити на основні й допоміжні. До основних процесів виконання можна віднести сам **процес виконання плану проекту**. Серед допоміжних процесів можна відзначити:

облік виконання – підготовку і розподіл необхідної для учасників проекту інформації з необхідною періодичністю;

підтвердження якості – регулярну оцінку виконання проекту з метою підтвердження відповідності прийнятим стандартам якості;

підготовку пропозицій – збір рекомендацій, відгуків, пропозицій, заявок і т. д.;

вибір постачальників – оцінку пропозицій, вибір постачальників та підрядників і укладання контрактів;

контроль контрактів – контроль за виконанням контрактів постачальниками й підрядниками;

розвиток команди проекту – підвищення кваліфікації учасників команди проекту.

4. Процеси аналізу.

Процеси аналізу включають аналіз плану й аналіз виконання проекту.

Аналіз плану означає визначення того, чи задовольняє складений план виконання проекту запропоновані до проекту вимоги й очікування учасників проекту. Він виражається в оцінці показників плану командою й іншими учасниками проекту.

На стадії планування результатом аналізу плану може бути ухвалене рішення про необхідність зміни початкових умов і складання нової версії плану або прийняття розробленої версії як базовий план проекту, що надалі є основою для виміру виконання. В управлінні проектуванням аналіз плану не виділяється в окрему групу, а включається в групу планування, роблячи цю групу за своєю природою ітеративною. Таким чином, під **процесами аналізу** маються на увазі процеси аналізу виконання.

Процеси аналізу виконання призначені для оцінки стану і прогнозу успішності виконання проекту відповідно до критеріїв і обмежень, визначених на стадії планування. Для більшості проектів до числа основних обмежень і критеріїв успіху входять цілі, терміни, якість і вартість робіт проекту. За негативного прогнозу ухвалюється рішення про необхідність коригувальних впливів, вибір яких здійснюється в процесах управління змінами.

Процеси аналізу також можна підрозділити на основні й допоміжні. До основних відносяться ті процеси аналізу, що безпосередньо пов'язані з цілями проекту і показниками, що характеризують успішність виконання проекту:

аналіз термінів – визначення відповідності фактичних і прогнозних термінів виконання операцій проекту директивним чи запланованим;

аналіз вартості – визначення відповідності фактичної і прогнозної вартості операцій і фаз проекту директивним чи запланованим;

аналіз якості – моніторинг результатів з метою їх перевірки на відповідність прийнятим стандартам якості і визначення шляхів усунення причин небажаних результатів виконання якості проекту;

підтвердження цілей – процес формального приймання результатів проекту його учасниками (інвесторами, споживачами і т. д.).

Допоміжні процеси аналізу пов'язані з аналізом факторів, що впливають на цілі й критерії успіху проекту. Ці процеси включають:

оцінку виконання – аналіз результатів роботи і розподіл проектної інформації з метою забезпечення учасників проекту даними про те, як використовуються ресурси для досягнення цілей проекту;

аналіз ресурсів – визначення відповідності фактичного і прогнозного завантаження і продуктивності ресурсів запланованим, а також аналіз відповідності фактичних витрат матеріалів, машинного часу і т. д. за планованим значенням.

До числа процесів аналізу не включений аналіз взаємодії з метою оптимізації процедур обробки проектної інформації, аналіз виконання контрактів з метою своєчасного внесення змін і запобігання суперечкам і ряду інших процесів, що не мають регулярного характеру (як аналіз взаємодії) або становлять частину включених процесів (як аналіз контрактів). У результаті аналізу ухвалюється рішення про продовження виконання проекту за окресленим раніше планом або визначається необхідність застосування коригувальних впливів.

5. Процеси оперативного управління.

Управління виконанням проекту – це визначення і застосування необхідних керівних впливів з метою успішної реалізації проекту. Якщо виконання проекту відбувається відповідно до наміченого плану, то управління фактично зводиться до виконання – доведення до учасників проекту планових завдань і контролю за їхньою реалізацією. Ці процеси включаються в процеси виконання.

У тому випадку, якщо в процесі реалізації виникли відхилення, аналіз яких показав, що необхідні визначення і застосування коригувальних впливів, потрібно:

- знайти оптимальні коригувальні впливи;
- скорегувати план робіт, що залишилися;
- погодити намічені зміни з усіма учасниками проекту.

Процеси оперативного управління призначаються для визначення, узгодження і внесення необхідних змін у план проекту. Такі процеси управління часто називаються **управлінням змінами** й ініціюються процесами аналізу.

До основних процесів оперативного управління, що зустрічається практично в кожному проекті, відносяться:

загальне управління змінами – визначення, узгодження, затвердження і прийняття до виконання коригувальних впливів і координація змін по всьому проекту;

управління ресурсами – внесення змін до складу і призначення ресурсів на роботи проекту;

управління цілями – коректування цілей проекту за результатами процесів аналізу;

управління якістю – розробка заходів щодо усунення причин незадовільного виконання.

Серед допоміжних процесів управління виділяють:

управління ризиками – реагування на події та зміну ризиків у процесі виконання проекту;

управління контрактами – координацію роботи субпідрядників, коректування контрактів, вирішення конфліктів.

6. Процеси завершення.

Завершення проекту супроводжується наступними процесами:

закриттям контрактів – завершенням і закриттям контрактів, включаючи вирішення усіх виниклих суперечок;

адміністративним завершенням – підготовкою, збором і перерозподілом інформації, необхідної для формального завершення проекту.

У разі реалізації всіх перерахованих вище процесів управління, що утворюють контур управління, використовуються визначені методи і засоби.

14.4. Зовнішні структури організації робіт із проектування ІС

Структури організації робіт із проектування мають бути розглянуті на двох рівнях: зовнішньому і внутрішньому.

Зовнішня структура відбиває певні зв'язки і відносини між окремими виконавцями і групами, залученими до виконання проекту, та їх материнськими підрозділами, відділами, компаніями.

Внутрішня структура відбиває стосунки між окремими виконавцями і групами, які виконують проект. Ці структури розглядаються незалежно від зовнішніх зв'язків. Внутрішня структура існує всередині зовнішньої, її варто розглядати для великих проектів.

Організація процесів розробки проекту ІС відрізняється значною складністю. До причин, що обумовлюють складність даних процесів, варто віднести, насамперед:

масштаби розробки ІС;

взаємозв'язок різних за своєю природою елементів проекту ІС (інформаційних, програмних і технічних засобів обробки інформації; економіко-математичних моделей; методів і засобів проектування; фахівців-розроб-

ників; елементів проекту системи та ін.);
різні фактори старіння зазначених елементів;
різний часовий цикл існування і темпів відновлення елементів;
тривалість процесу проектування системи;
індивідуальність проекту, обумовлену специфікою об'єкта проектування;

колективний характер праці багатьох фахівців різної кваліфікації.

Організація робіт із проектування ІС визначається порядком взаємодії між декількома сторонами, що беруть участь у ньому: користувачем, замовником, адміністратором і розробником.

Користувач (user) – це особа або організація, що використовує діючу систему для виконання заданої функції. Для ІС під користувачем розуміють, насамперед, адміністративно-управлінський апарат, для якого створюється ця система. Користувач виконує наступні функції:

формує вхідні дані для проектування й обробки;

визначає склад завдань для автоматизації;

визначає основні вимоги до завдань і режими функціонування системи.

Замовник (customer) – це організація, що замовляє або отримує систему, програмний продукт чи послугу від постачальника.

Замовник виконує наступні функції:

формує вимоги до системи і її частин;

видає технічне завдання, фінансує розробку ІС;

забезпечує проведення комплексу заходів щодо її створення;

проводить упровадження і прийом проекту ІС.

При цьому замовник відповідає перед користувачем за відповідність складу і характеристик розв'язуваних завдань, режимів функціонування ІС вхідним даним користувача, за терміни створення системи, правильність використання ресурсів у процесі проектування.

Оператор (operator) – організація, що експлуатує систему (оперує із системою, керує системою). Оператор відповідає перед користувачем за правильність і своєчасність результатів роботи ІС, а перед замовником і розробником – за дотримання умов експлуатації, вимог до технічної документації.

Розробник (developer) – це організація, що провадить дії з розробки (включно з аналізом вимог, проектуванням, випробуваннями під час приймання) в межах життєвого циклу програмного забезпечення. Розробник виконує наступні функції:

розробляє ІС за технічним завданням замовника;

бере участь у впровадженні;

здійснює задачу проекту замовнику;
здійснює авторський супровід проекту.

Розробник відповідає перед замовником за правильність реалізації вимог ТЗ на ІС, науково-технічний рівень розробки, терміни проведення робіт, якість проектної документації, правильність витрати грошових ресурсів. Під розробником мається на увазі як одна організація, так і певна сукупність організацій, у яку входять головна організація й організації-співвиконавці.

Існують різні типи організації робіт за участю чотирьох сторін, вибір якого залежить від обсягу замовлення.

Якщо замовлення має невеликі розміри за вартістю і за тривалістю робіт, то приймають схему, в якій в одній особі виступають замовник, розробник і оператор (рис. 70).

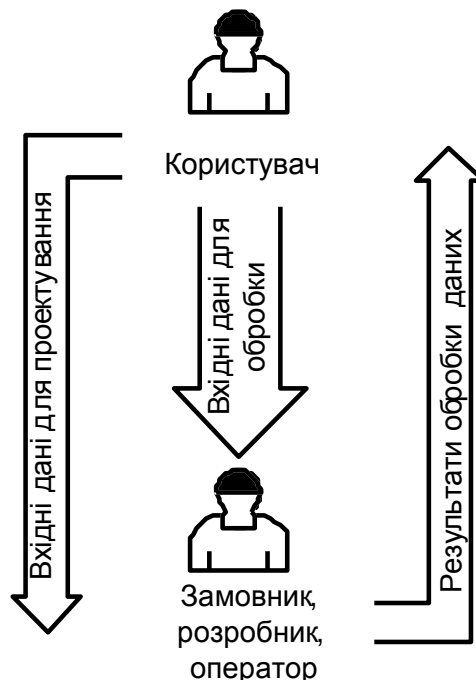


Рис. 70. Структура організації робіт для невеликих замовлень

Переваги даної структури:

мінімальна кількість організацій-учасників процесу;
мінімальні терміни і вартість розробки.

Недоліки:

відсутній дієвий контроль за науково-технічним рівнем розробки, термінами виконання робіт;

недосягнення високого професійного рівня розробників.

Ці недоліки обумовлені поєднанням в одній організації функцій сторони, що розробляє, і сторони, що приймає.

2. Для великих і складних замовлень застосовують структуру, відповідно до якої функції розробника відокремлюються від функцій

замовника й адміністратора і виконуються іншою організацією (рис. 71).

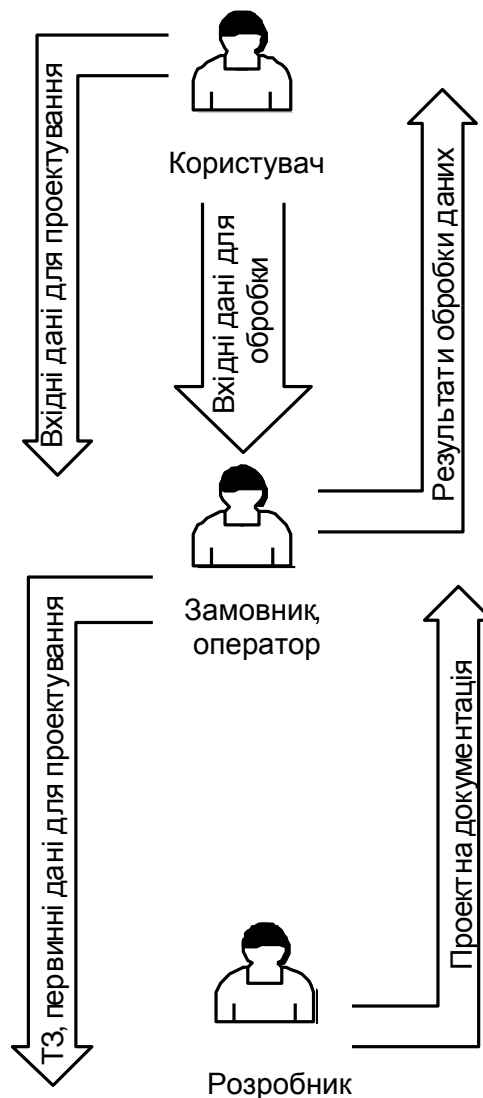


Рис. 71. Структура організації робіт при відокремленні замовника від розробника

Переваги даної структури:

раціональний розподіл функцій між сторонами, що беруть участь у створенні й експлуатації ІС;

можливість залучення до розробки ІС організацій, що спеціалізуються на розробці і впровадженні таких систем.

Недоліки:

відсутність прямого зв'язку між розробником і користувачем, що створює труднощі у своєчасному одержанні й деталізації вихідних даних для проектування;

певні труднощі при прийомі проекту до експлуатації через бажання операторів одержати методологічне забезпечення завдань, які

максимально відповідають ідеальним умовам експлуатації, що, у свою чергу, вимагає великих термінів і обсягів з доробки проекту.

3. У тому випадку, якщо замовник – велика організація, що курирує розробку кількох проектів ІС, застосовують наступну структуру (рис. 72).

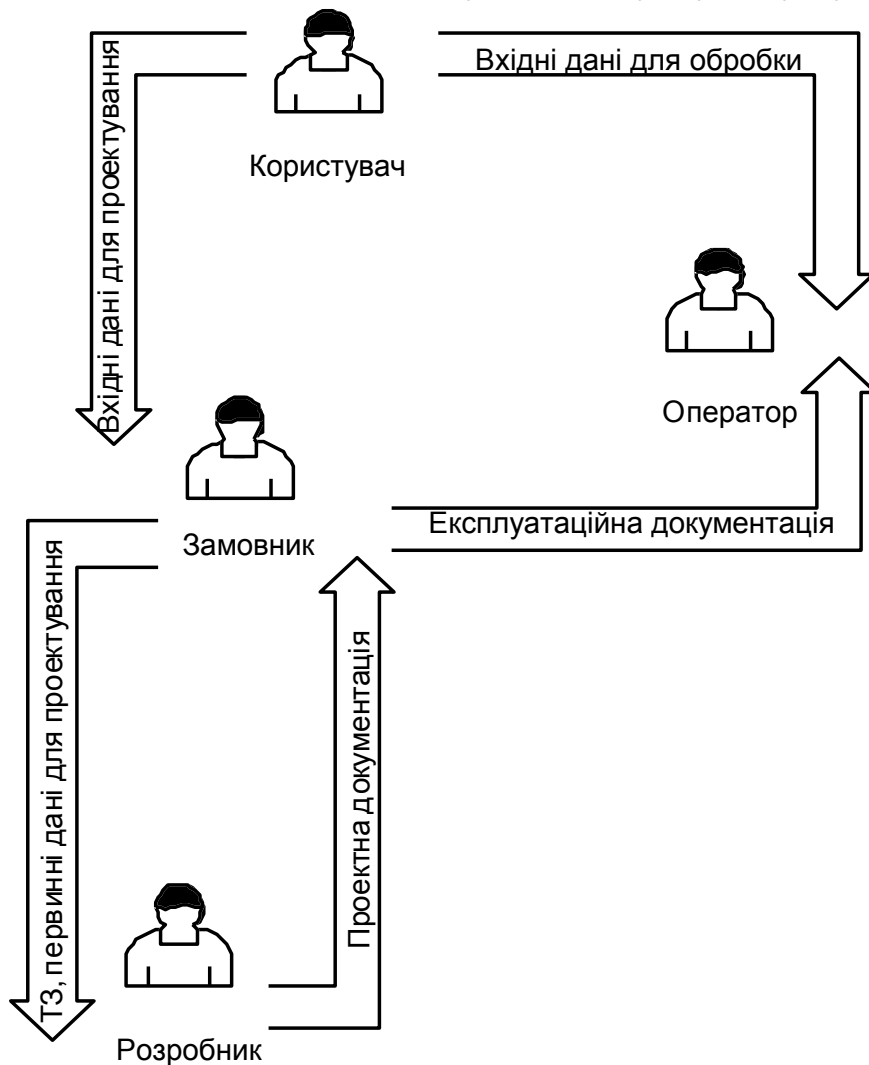


Рис. 72. Структура організації робіт при повному поділі функцій сторін

Дана структура характеризується тим, що на замовника покладаються функції супроводу, замовлення і приймання проектів кількох ІС.

Переваги даної структури:

більш високий ступінь спеціалізації працівників, отже, більш високий професійний рівень;

можливість організації контролю за термінами і якістю виконання робіт.

14.5. Організаційні форми управління проектами

Управління проектами, як правило, розглядають у двох аспектах:

організаційному, за якого управління проектом розглядається за

рівнями організаційно-адміністративної структури з відповідними правами й обов'язками суб'єктів процесу проектування;

функціональному, як застосування відповідних механізмів (методів і засобів) організації та ведення проектних робіт.

На практиці дані аспекти управління проектами реалізуються в конкретних формах управління, застосовуваних в організаціях-розробниках систем.

У загальному випадку організаційна структура управління проектом регулює взаємини підрозділів і посадових осіб в організації, встановлює розподіл ролей, повноважень і відповідальності між ними, а також порядок функціонально-технічних зв'язків, що виникають у процесах управління. Організаційна структура й організаційний механізм як система зв'язку в даній організації у всьому різноманітті проявів утворюють **організаційні форми управління** діяльністю колективу.

Форми управління, застосовувані в організаціях-розробниках ІС, залежать від виконуваних робіт. Як правило, в організаціях-розробниках виконуються роботи, пов'язані з проектуванням ІС, з підтримкою і супроводом ІС.

Організаційна форма управління проектуванням ІС відіграє велику роль у реалізації завдань підвищення ефективності процесу розробки систем. Форма управління є тим стрижнем, що багато в чому визначає зміст і якість проекту системи. Можна передати в розпорядження розробників найдосконаліші засоби проектування, чіткі форми документації, плани робіт, методи контролю, але без належної організації не одержати проекту, що задовольняє потреби замовника. І навпаки, досконала форма організації проектування заповнює нестачу ефективних засобів проектування і в окремих випадках – навіть кваліфікації розробників.

Організаційні форми управління в організаціях – розробниках ІС здійснюються за функціональним, проектним (цільовим) і матричним принципами.

Функціональний принцип побудови структури організації використовується в процесі виконання завдань проектування постійного характеру.

Для виконання кожного виду завдань, наприклад розробки постановки економічних завдань, інформаційного забезпечення і т. д., формуються функціональні підрозділи з фахівців визначеного профілю. Подібна організаційна структура має високий ступінь централізації управління, притаманний їй авторитарний стиль керівництва. У галузі розробки ІС функціональна структура організації зустрічається дуже рідко.

За використання **проектного принципу** формується організаційний підрозділ – проектна група (проект), що призначений для одноразової розробки ІС. Фахівці проектної групи утворюють автономну організаційну одиницю, керівник (головний конструктор) якої має відповідні повноваження і несе повну відповідальність за результати діяльності проектної колективу, який після виконання проекту може бути розформований. Цей принцип найбільш часто використовується для побудови організаційних структур проектних організацій.

Матрична побудова організаційних структур передбачає формування в організації-розробнику ІС із фахівців функціональних підрозділів проектних груп для розробки конкретних проектів. При цьому фахівці не втрачають належності до відповідного функціонального підрозділу і перебувають у подвійному підпорядкуванні: у керівника проекту (відповідальність за проектом) і в керівника функціонального підрозділу (організаційна відповідальність).

Матричні структури застосовуються в умовах високого ступеня кооперації функціональних підрозділів. Ці структури засновані на особливому механізмі взаємодії функціональних і проектно-цільових підсистем апарата управління проектною організацією. Головна особливість матричних структур полягає в обов'язковому виділенні конкретної особи – керівника проекту, наділеного всією повнотою відповідальності за досягнення мети проектування і значних розпорядницьких прав, що делегуються йому вищим керівництвом.

За одночасної розробки кількох проектів в організаційну структуру, як правило, вводиться планово-виробничий підрозділ, головним завданням якого є балансування ресурсів, споживаних усіма проектами, і забезпечення координації та поточних змін у проектах.

14.6. Поділ праці у проектних колективах

Проект ІС складається з різномірних елементів: інформаційних, програмних, технічних, функціональних. Процес проектування складається з багатьох етапів, на кожному з яких вирішуються різні завдання, його реалізація вимагає різних знань, у ньому беруть участь фахівці різних профілів і кваліфікації. Тому існує об'єктивна потреба в поділі праці в колективі розробників системи.

Поділ праці, як відомо, сприяє підвищенню продуктивності праці, у тому числі й проектувальників ІС; накопиченню досвіду і знань; підвищенню якості проектних рішень; умінню вирішувати складні проблеми на рівні фахівця середньої кваліфікації. Безпосередньо з поділом праці пов'язана проблема чисельності в проектній групі.

У проектних колективах поділ праці між фахівцями здійснюється, як правило, на основі одного з двох наступних принципів: поопераційного (технологічного) чи підсистемного.

Поділ праці на **поопераційній (технологічній)** основі базується на властивості декомпованості процесу проектування ІС на технологічні операції, що виконуються окремими фахівцями чи групами фахівців. У цьому випадку потрібна чітка регламентація інтерфейсів між операціями. З цим пов'язані високі вимоги до документування проекту. Зв'язок між поділом праці і накопиченням спеціальних знань і досвіду приводить до професійної спеціалізації розробників ІС (наприклад, експерт з інформаційних технологій, системний аналітик, програміст, оператор і т. д.).

Поділ праці в колективі розробників ІС на основі поопераційного принципу, як правило, складний з огляду на наступні фактори:

невисокий рівень типізації технологічних операцій проектування ІС;
неможливість одержання об'єктивно точної якісної оцінки проміжних результатів проектування;

відсутність об'єктивних критеріїв нормування праці фахівців;

низький ступінь стандартизації й уніфікації компонентів ІС.

Підсистемний поділ праці в колективі розробників ІС базується на властивості декомпованості проекту на підсистеми, кожна з яких незалежно від кількості технологічних операцій проектування розробляється окремою групою фахівців. У цьому випадку передбачаються стандартизація й уніфікація інтерфейсів між підсистемами на кожному етапі процесу проектування ІС. Накопичення знань і досвіду приводить до системної спеціалізації розробників ІС (наприклад, фахівців з інформаційного забезпечення, технічного забезпечення, експертних систем і т. д.) або до спеціалізації з розробки компонентів ІС (інформаційної бази, інтерфейсу користувача і т. д.).

На практиці під час поділу праці в проектних колективах можливе використання обох вищезгаданих принципів.

14.7. Функціональні ролі в колективі розробників

Функції, що виконуються розробниками, в проекті підрозділяються на: **організаційні**, які створюють умови для виконання проектних завдань; **виробничі**, які безпосередньо пов'язані з цими завданнями.

У концепції Microsoft Solution Framework визначено шість рольових кластерів, які відповідним чином структурують проектні функції розробників (рис. 73). Вони відповідальні за різні сфери компетенції

і партнерам. Ключовим моментом є чітке визначення працівників, відповідальних за кожен рольовий кластер, їх функцій, відповідальності й очікуваного внеску в кінцевий результат.

Таблиця 28

Характеристика рольових кластерів проектної групи MSF

Рольовий кластер 1	Мета 2	Область компетенції 3	Функції 4
Управління продуктом	Задоволення замовників	Маркетинг. Бізнес-віддача (бізнес-пріоритети). Подання інтересів замовника. Планування продукту	Виступає в ролі представника замовника. Формує спільне бачення/рамки проекту. Організовує роботу з вимогами замовника. Розвиває сфери застосування в бізнесі. Формує очікування замовника. Визначає компроміси між параметрами "можливості продукту / час / ресурси". Організовує маркетинг та PR. Розробляє, підтримує і виконує план комунікацій
Управління програмою	Досягнення результату в рамках проектних обмежень	Управління проектом. Вироблення архітектури рішення. Контроль виробничого процесу. Адміністративні служби	Управляє процесом розробки з метою отримання готового продукту у відведені терміни. Формулює специфікацію продукту і розробляє його архітектуру. Регулює взаємини і комунікацію всередині проектної групи. Стежить за часовим графіком проекту і готує звітність про його стан. Проводить в життя важливі компромісні рішення. Розробляє, підтримує і виконує зв'язаний план і календарний графік проекту

1	2	3	4
Розробка	Створення продукту відповідно до специфікації	Технологічне консультування. Проектування і здійснення реалізації. Розробка застосувань. Розробка інфраструктури	Визначає деталі фізичного дизайну. Оцінює необхідні час і ресурси на реалізацію кожного елемента дизайну. Розробляє або контролює розробку елементів. Готує продукт до впровадження. Консультує команду з технологічних питань
Тестування	Схвалення випуску продукту тільки після того, як усі дефекти виявлені і усунуті	Планування тестів. Розробка тестів. Звітність за тестами	Забезпечує виявлення всіх дефектів. Розробляє стратегію і плани тестування. Здійснює тестування
Задоволення споживача	Підвищення ефективності користувача, збільшення споживчої цінності продукту	Забезпечення технічної підтримки. Навчання. Ергономіка. Графічний дизайн. Інтернаціоналізація. Загальнодоступність	Подає інтереси споживача в команді. Організовує роботу з вимогами користувача. Проектує і розробляє системи підтримки продуктивності. Визначає компроміси, що відносяться до зручності використання і споживчих якостей продукту
Управління випуском	Безпроблемне впровадження і супровід продукту	Інфраструктура. Супровід. Бізнес-процеси. Управління випуском готового продукту	Подає інтереси відділів постачання і обслуговування продукту. Організовує постачання проектної групи. Організовує впровадження продукту. Виробляє компроміси в керованості та зручності супроводу продукту. Організовує супровід і інфраструктуру постачання. Організовує логістичне забезпечення проектної групи

У компанії IBM визначена своя рольова структура проекту. Ця структура включає достатньо повний перелік типових ролей, узгоджений з багатьма реальними дисциплінами розвитку програмних проектів. Водночас вона представляє ролі розробників в організаційному контексті, тобто розглядає не лише розробників, але й тих, хто, не беручи участі в проекті як виконавці, здійснює вплив на постановку завдань проекту, на виділення ресурсів і забезпечення здійсненності розвитку робіт. У наведеному переліку характеристика кожної ролі, за

суті, задає коло споріднених організаційних і виробничих функцій, які об'єднуються з метою визначити роль.

Замовник (Customer) – реально існуючий (у організації, якій підпорядкована команда) ініціатор розробки або хто-небудь інший, уповноважений приймати результати (як поточні, так і остаточні) розробки.

Планувальник ресурсів (Planner) – висуває і координує вимоги до проектів в організації, що здійснює дану розробку, а також розвиває і направляє план виконання проекту з погляду організації.

Менеджер проекту (Project Manager) – відповідає за розвиток проекту в цілому, гарантує, що розподіл завдань і ресурсів дозволяє виконати проект, що роботи і подання результатів йдуть за графіком, що результати відповідають вимогам. У рамках цих функцій менеджер проекту взаємодіє із замовником і планувальником ресурсів.

Керівник команди (Team Leader) – здійснює технічне керівництво командою в процесі виконання проекту.

Архітектор (Architect) – відповідає за проектування архітектури системи, погоджує розвиток робіт, пов'язаних із проектом.

Проектувальник підсистеми (Designer) – відповідає за проектування підсистеми або категорії класів, визначає реалізацію й інтерфейси з іншими підсистемами.

Експерт предметної сфери (Domain Expert) – відповідає за вивчення сфери застосування, підтримує спрямованість проекту на вирішення завдань даної сфери.

Розробник (Developer) – реалізує проєктовані компоненти, володіє і створює специфічні класи та методи, здійснює кодування і автономне тестування, будує продукт. Це широке заняття, яке може підрозділятися на спеціальні ролі (наприклад, розробник класів). Залежно від складності проекту команда може включати різну кількість розробників.

Розробник інформаційної підтримки (Information Developer) – створює документацію, яка супроводжує продукт, коли випускається версія. Інсталяційні матеріали, що включаються в неї, так само, як і посилальні й навчальні, а також матеріали допомоги, надаються на паперових і машинних носіях. Для складних проектів можливий розподіл цих завдань між декількома розробниками інформаційної підтримки.

Фахівець з призначеного для користувача інтерфейсу (Human Factors Engineer) – відповідає за зручність застосування системи.

Працює із замовником, щоб упевнитися, що призначений для користувача інтерфейс задовольняє вимоги.

Тестувальник (Tester) – перевіряє функціональність, якість і ефективність продукту. Будує і виконує тести для кожної фази розвитку проекту.

Бібліотекар (Librarian) – відповідає за створення і ведення спільної бібліотеки проекту, яка містить всі проектні робочі продукти, а також за відповідність робочих продуктів стандартам.

З перелічених ролей замовник і планувальник ресурсів мають лише зовнішнє відношення до розробки проекту – вони не є членами команди. Замовник – це особа, зацікавлена в отриманні результатів. Планувальник вирішує завдання розподілу фінансових, трудових і технічних ресурсів для різних проектів всередині фірми. За правильної організації розробки з цими дійовими особами доводиться стикатися лише менеджерів проекту.

Незважаючи на різні підходи до вибору ролей, можна виділити перелік ключових ролей, що характеризують найбільш типові ситуації для програмних проектів:

- архітектор проекту;
- проектувальники підсистем;
- керівники команд розробки підсистем;
- фахівець з призначеного для користувача інтерфейсу;
- експерт предметної сфери.

14.8. Внутрішні організаційні структури проектної групи

Вибір доцільного поділу праці розробників ІС залежить від ряду факторів, що впливають різною мірою на вирішення проблеми. Найбільш істотними факторами є наступні:

- потенціал колективу розробників;
- обсяг і складність розроблювальних проектів;
- технологія проектування системи;
- модель життєвого циклу системи.

Ступінь впливу кожного фактора в конкретних випадках приводить до великої розмаїтості поділу праці і пов'язаних з ним організаційних форм управління проектуванням ІС у проектній групі. При цьому використовуються, як правило, наступні типові організаційні структури проектної групи: відкрита, централізована, децентралізована, великого проекту.

Відкрита організаційна структура відрізняється тим, що

закріпленого організаційного розподілу обов'язків немає. Кожен член колективу розробників є неформальним керівником на етапі розробки системи, де він кваліфікованіший, ніж інші. Обов'язки на окремих етапах розподіляються між розробниками відповідно до їхніх знань, досвіду і здібностей.

Адміністративний керівник у групі здійснює, як правило, наступні функції:

взаємодію з замовником;

планування і контроль термінів;

розподіл ресурсів, координацію робіт;

звітність перед керівництвом організації (якщо група працює в складі такої).

Така організаційна структура формується з 7 – 10 осіб для творчого вирішення завдань і рекомендується для робіт, виконуваних на ранніх етапах проектування системи, – проведення обстеження предметної сфери (об'єкта управління), аналізу й розробки концепції проекту. Така чисельність проектувальників дає можливість повного обміну інформацією між ними, а також мати відносно невисокі витрати на адміністрування. Відкрита організаційна структура дозволяє варіювати кількість розробників, залучаючи до виконання робіт найбільш кваліфікованих фахівців, що сприяє підвищенню якості проекту.

Централізована організаційна структура проектної групи передбачає в якості керівника фахівця високої кваліфікації, що здійснює адміністративне і технічне керівництво. Він же є основним посередником між групою, замовником проекту і зовнішніми організаціями.

Дана структура найбільш прийнятна для вирішення завдань, що мають жорсткі обмеження за термінами і витратами, на розробку системи. Чисельність такої групи – до 7 чоловік. Особливістю даної організаційної структури проектної групи є чіткий розподіл функцій і повноважень між фахівцями. Результати роботи кожного члена групи даються в розпорядження всіх учасників процесу проектування. Недолік полягає у відсутності прояву ініціативи конкретними виконавцями. Прикладом централізованої структури може слугувати група головного фахівця. У такій ролі може виступати головний конструктор проекту. У групу входять головний фахівець, його заступник, аналітики (постановники завдань), програмісти й адміністратор.

Головний фахівець виконує наступні функції:

відповідає за розробку загальної концепції проекрованої ІС і відповідність проектних рішень вимогам користувача;

виконує разом з аналітиками декомпозицію системи;

контролює терміни проектування і повноту проектної документації; несе відповідальність за розробку проекту у всіх аспектах.

Головний фахівець здійснює безпосереднє управління проектом і визначає стратегію проектування.

Заступник головного фахівця орієнтований на тактичні питання проектування ІС, на аналіз альтернатив у розробці проектних рішень. Заступник перебуває в курсі всіх питань проекту й у будь-який необхідний момент може взяти на себе роль керівника проектної групи. Він же бере безпосередню участь у розробці проекту.

Аналітики і програмісти здійснюють безпосередньо розробку частин проекту.

Бібліотекар здійснює організацію і ведення бібліотеки проектних рішень чи словника даних (опис постановок завдань, програм і т. д.). На ньому лежить відповідальність за використання наявних ресурсів у процесі проектування ІС.

Таким чином, група головного фахівця характеризується концентрацією відповідальності й основних завдань проекту в руках одного фахівця і звільненням розробників від адміністративної й організаційної роботи.

Перевагами даної організації праці проектувальників ІС є застосування спадного ("зверху-вниз") проектування, підвищення продуктивності праці проектувальників, підвищення якості проектних рішень, інтенсивне навчання й ефективне використання початківців-розробників ІС. Слід також зазначити, що в даному випадку висувуються високі вимоги до кваліфікації й організаторських здібностей головного фахівця.

Головний фахівець проекту спільно з фахівцями планово-виробничого відділу здійснює планування й оперативний контроль проектних робіт на всіх етапах технологічного процесу проектування системи. У його компетенцію входить складання поопераційного плану – основного документа з розробці ІС, а також маневрування ресурсами проектування в рамках своєї групи.

Децентралізована організаційна структура проектної групи має властивість двох вищевикладених структур. Дана організаційна структура застосовується в колективах з великою чисельністю розробників (понад 10 чоловік), що здійснюють проектування великих ІС, які декомпонуються на підсистеми (контури, модулі) і комплекси завдань.

У цьому випадку **керівник проекту** здійснює управління групою старших фахівців, що відповідають за розробку великих частин системи, а ті, у свою чергу, здійснюють керівництво молодшими спеціалістами, що під-

тримують між собою горизонтальні зв'язки в процесі проектування. Як правило, молодші спеціалісти об'єднані в підгрупи за технологічною спеціалізацією. Ця організаційна структура має багато спільного з групою головного фахівця, але з додаванням нових посад, що розширюють можливості групи розробників з проектування більших систем.

Керівник проекту здійснює організаційне й методичне керівництво головними фахівцями частин проекту. Такий розподіл ролей забезпечує концептуальну єдність великих проектів. При цьому головною особливістю є робота керівника проекту, що полягає в тому, щоб використовувати свої професійні технічні знання винятково в інтересах користувача.

Організаційна структура на основі так званої **моделі великого проекту**, що застосовується на практиці в різних варіантах для розробки великих проектів. У цьому випадку формуються проектні колективи до 40 осіб. Особливістю даної організаційної структури є виділення, як мінімум, чотирьох груп фахівців у складі колективу розробників.

При цьому виділяються наступні групи фахівців:

група системного аналізу і проектування;

група реалізації (група програмування);

група тестування;

адміністративна група.

Група системного аналізу і проектування виконує системний аналіз, розробляє специфікації до автоматизованої системи, здійснює проектування ІС, під час реалізації проекту виконує контрольні функції і документування системи відповідно до вимог стандартів. **Група реалізації** здійснює розробку робочого проекту. Таким чином, ядро групи системного аналізу і проектування зберігається і на етапі реалізації (Техноробочого проектування), на якому фахівці групи системного аналізу і проектування очолюють групи програмістів. Поділ праці в групі системного аналізу і проектування, як правило, здійснюється на основі **поопераційного** принципу, а в групі реалізації – на основі **підсистемного** принципу.

Важливим в організації процесу проектування ІС у моделі великого проекту є формування **групи тестування**, що проводить комплексне налагодження і приймальні іспити системи. **Адміністративна група** виконує технічні й адміністративні функції, включаючи підготовку проектної документації, контроль документації, звітності і т. д.

Варто зауважити, що для розробки ІС до складу проектної групи конкретної організації можуть залучатися на тимчасове співробітництво фахівці з боку користувачів і фахівці-розробники для вирішення специфічних завдань, що вимагають високої кваліфікації і практичного досвіду в конкретній предметній сфері. Участь з боку користувача доцільна на етапі

системного аналізу і розробки вимог до системи, а також під час проектування інтерфейсу користувача.

Незалежно від форми організаційної структури колективу розробників для забезпечення процесів проектування повинен бути створений спеціальний підрозділ, що організує використання комп'ютерів, автоматизованих робочих місць проектувальників, термінальних станцій і т. д.; супровід базового програмного забезпечення; поточний ремонт технічних засобів.

14.9. Організаційні форми реінжинірингу бізнес-процесів

Реінжиніринг бізнес-процесів у вузькому розумінні передує проектуванню ІС і відповідає в традиційному уявленні етапу системного аналізу, на якому, зокрема, визначаються вимоги до інформаційної системи. Разом з тим завдання нестандартного виділення бізнес-процесів, що забезпечують різке підвищення ефективності діяльності підприємства, обумовлюють розробку нових організаційних форм участі фахівців у роботах з реінжинірингу.

Сам по собі реінжиніринг бізнес-процесів становить творчу діяльність колективів фахівців, що виробляють концепцію здійснення бізнес-процесів з використанням передових інформаційних технологій. До таких колективів висуваються наступні вимоги.

1. Участь представників вищої ланки управління чи лідерів проекту на всіх етапах реінжинірингу бізнес-процесів, що відповідають за вироблення і реалізацію рішень з реінжинірингу бізнес-процесів.

2. Спільна робота користувачів (власників бізнес-процесів) і груп (команд) реінжинірингу з визначення змісту перепроєктованих бізнес-процесів.

3. Включення сторонніх консультантів до складу груп реінжинірингу на рівних правах із працівниками підприємства.

4. Міждисциплінарний характер складу учасників груп реінжинірингу, що включають представників усіх підрозділів, що взаємодіють у бізнес-процесі.

5. Відволікання учасників груп реінжинірингу від основної роботи на період реінжинірингу бізнес-процесів.

6. Контроль з боку груп реінжинірингу реалізації і впровадження сформованого проекту.

7. Координація роботи груп реінжинірингу з кількох бізнес-процесів на рівні методологічних центрів.

8. Безупинне планування і контроль робіт з реінжинірингу бізнес-процесів з боку адміністрації підприємства.

Організаційна структура проекту реінжинірингу бізнес-процесів наведена на рис. 74.

Лідер проекту – це менеджер верхньої ланки управління, що наділений повноваженнями очолювати роботи з реінжинірингу бізнес-процесів на всіх його етапах. Лідер проекту повною мірою усвідомлює завдання реінжинірингу, їхню складність і трудомісткість, і багато в чому завдяки його енергії забезпечується успіх проекту. Тому лідер проекту не може бути формально призначений без усвідомлення ним необхідності і можливості реінжинірингу, а також здатності переключити основний робочий час на проведення реорганізації підприємства.

Найважливіші вимоги, які висуваються до лідера проекту, полягають, з одного боку, у здатності концептуально мислити, узагальнюючи моделі всіх бізнес-процесів, що реорганізуються, а, з іншого боку, у здатності мотивувати проведення робіт з реінжинірингу бізнес-процесів з боку всіх його учасників.



Рис. 74. Організаційна структура проекту з реінжинірингу бізнес-процесів

Основні методи роботи лідера проекту полягають у наступних заходах: проведенні зборів з колективом працівників підприємства, на яких ставляться завдання реінжинірингу, пояснюється роль співробітників і їхня участь у реінжинірингу бізнес-процесів;

підборі й контролі за роботою кадрів (власників процесу, груп реінжинірингу, методологічних центрів);

проведенні координаційних нарад за ходу проведення реінжинірингу бізнес-процесів;

участі в засіданнях керівного комітету на стадіях планування, контролю і приймання робіт з реінжинірингу бізнес-процесів.

Власники бізнес-процесів – це майбутні адміністратори процесів, що на стадії реінжинірингу виконують забезпечення проекту в рамках кожного окремого бізнесу-процесу всіма необхідними ресурсами. Звичайно власники бізнес-процесів призначаються лідером проекту з кількості керівників лінійних чи функціональних підрозділів, на базі яких надалі передбачається організація процесних підрозділів (груп).

Роль власника бізнес-процесу полягає в тому, щоб бути посередником між групою реінжинірингу і колективом підприємства, що продовжує функціонувати. Зокрема, власник бізнес-процесу забезпечує членів групи реінжинірингу всією необхідною інформацією про функціонування підприємства і пояснює працівникам підприємства сутність змінюваної технології роботи.

Команди (групи) реінжинірингу БП розробляють проект і виконують реінжиніринг бізнес-процесів. Кількість груп реінжинірингу відповідає кількості бізнес-процесів, що реорганізуються. До складу груп реінжинірингу входять **інсайдери** – працівники підприємств і їхніх структурних підрозділів, що беруть участь у бізнес-процесі, а також **аутсайдери**, працівники загальносистемних підрозділів, наприклад відділів інформаційних технологій, планування або зовнішні консультанти. Серед членів групи реінжинірингу вибирається капітан, що координує виконання проектної роботи.

У групі реінжинірингу рекомендується співвідношення інсайдерів до аутсайдерів у співвідношенні 3:1 при загальній чисельності групи 7 – 10 осіб. Особливість роботи групи реінжинірингу полягає в активній участі всіх членів групи, тобто недопущенні ситуації, коли аутсайдери активні, а інсайдери пасивні (аутсайдери задають питання інсайдерам і формують проектні рішення) чи, навпаки, інсайдери активні, а аутсайдери пасивні (інсайдери задають питання аутсайдерам, як чинити в тій чи іншій ситуації).

Основний метод роботи групи реінжинірингу полягає в колективному виробленні проектних рішень шляхом генерації пропозицій і їхнього критичного обговорення. При цьому роль аутсайдерів полягає в методологічному забезпеченні процесу реінжинірингу на основі методик і досвіду реінжинірингу в аналогічних сферах діяльності.

Для формалізації і вибору проектних рішень використовуються методи структурного, функціонально-вартісного і динамічного

моделювання й аналізу бізнес-процесів.

На стадії реалізації проекту реінжинірингу бізнес-процесів, зокрема в процесі створення інформаційної системи, група реінжинірингу контролює виконання робіт з боку розробників інформаційної системи і, таким чином, бере на себе функції **замовника системи**.

На стадії впровадження група реінжинірингу виконує функції **зовнішнього тестування** інформаційної системи і навчання членів процесних груп роботі в нових умовах.

Методологічний центр ("цар" РБП) координує роботу множини груп реінжинірингу і забезпечує їх методологією, інструментарієм, типовими рішеннями і звичайно формується з представників загальносистемних підрозділів підприємства (відділів інформаційних технологій, планування) або консалтингової фірми.

Методологічний центр формується лідером проекту і підкоряється останньому. Якщо лідер проекту здійснює періодичне керівництво проектом реінжинірингу, то методологічний центр виконує безперервне оперативне керівництво проектом. **Мета** роботи методологічного центру полягає в пошуку і реалізації загальносистемних рішень у частині стикування різних бізнес-процесів і розподілу загальних ресурсів на стадії функціонування.

Основний метод роботи методологічного центру полягає в аналізі виконання бізнес-процесів, що реорганізуються, на основі їх моделювання на стадії розробки проекту і вивчення статистики, одержуваної з функціональної інформаційної системи, на стадії впровадження бізнес-процесів.

Усі суперечності в бізнес-процесах, що реорганізуються, усуваються шляхом проведення координаційних нарад на рівні власників процесів і груп реінжинірингу.

Керівний комітет виділяє ресурси на підприємстві для проведення реінжинірингу і контролює виконання всіх етапів відповідно до розробленого плану-графіка робіт. Таким чином, керівний комітет – це контрольно-розпорядницький орган, очолюваний лідером проекту.

14.10. Сучасний ринок програмних проектів

Перехід економіки країни на ринкові відносини призвів до того, що в галузі інформаційних систем з'явився самостійний ринок послуг з проектування, до купівлі і встановлення обчислювальної техніки, розробки локальних мереж, прокладання мережного устаткування і

навчання користувачів, виконуваних компаніями – системними інтеграторами або компаніями – проектними інтеграторами.

14.10.1. Системна інтеграція

Залежно від спеціалізації діяльності виділяють наступні **види системних інтеграторів**:

мережні інтегратори – компанії, які спеціалізуються на мережних і телекомунікаційних рішеннях, які мають, у свою чергу, мережу своїх реселерів (продавців);

програмні інтегратори – компанії, які спеціалізуються на програмному забезпеченні;

комплексні системні інтегратори – компанії, які виконують комплексне вирішення завдань замовника в процесі побудови ІС, оскільки замовник готовий перекласти детальну проробку й реалізацію проекту на плечі системного інтегратора, визначивши лише вихідні дані та завдання, які повинна вирішувати реалізована ІС.

Останні, як правило, виконують наступний **набір функцій**:

продаж (дистрибуцію, постачання для проектів) апаратного забезпечення;

продаж (дистрибуцію, постачання для проектів) програмного забезпечення;

консалтинг, проектні роботи, сервіс, технічну підтримку, навчання.

За **структурою і виконуваними функціями** виділяють дві групи фірм системних інтеграторів:

малі фірми з кількістю співробітників до 50 осіб, що становлять невеликі динамічні компанії, які спеціалізуються на інтеграції програмного забезпечення й апаратних засобів кількох бізнесів-партнерів;

середні фірми з кількістю співробітників від 50 осіб до 100 і **великі фірми-інтегратори** з кількістю співробітників понад 100 осіб, що пропонують клієнту широкий спектр рішень, заснованих на устаткуванні великої кількості провідних закордонних виробників. У таких компаніях створені відділи розробки власних програмних продуктів, пропонуються послуги з консалтингу і навчання фахівців у власних навчальних центрах.

Об'єктами, з якими працюють фахівці фірм-інтеграторів, є:

офісні та корпоративні мережі;
багаторівневі системи збереження інформації;
системи управління технологічними процесами;
корпоративні автоматизовані інформаційні системи для великих підприємств: банків, нафтогазових компаній, великих хімічних комбінатів і т. д.

Можна виділити п'ять типів рівнів рішень, які забезпечуються ринком програмних проектів:

придбання окремих модулів програмно-технічних засобів у вже сформованих каналах розповсюдження комп'ютерної техніки і самостійна побудова конфігурації ІС;

звернення до підприємств – системних інтеграторів – для надання консультаційних послуг з вибору і придбання модулів програмно-технічних засобів, самостійна побудова ІС;

у ході створення великих комплексних проектів підприємство – системний інтегратор – надає консультаційні послуги з вибору, придбання і освоєння модулів програмно-технічних засобів, а також здійснює безпосередньо побудову конфігурації ІС;

підприємство – системний інтегратор не тільки створює ІС, але й бере на себе сервіс щодо усунення всіх неполадок у роботі програмно-технічних засобів у перебігу періоду часу, погодженого із замовником;

виконання проектів ІС і надання послуг з обслуговування програмно-технічних засобів незалежними спеціалізованими організаціями і персоналом, що найефективніше вирішують загальні та окремі інформаційні проблеми (аутсорсинг).

Термін "аутсорсинг" походить від англійського *outsourcing* і означає буквально "відкачування засобів". **Аутсорсинг** – використання зовнішніх ресурсів (матеріальних, трудових, інтелектуальних, інформаційних); відмова від власного бізнес-процесу і придбання послуг з реалізації цього бізнес-процесу в іншій організації. За аутсорсингу сторонній організації може бути доручене ведення бухгалтерії, підбір персоналу, підтримка корпоративної інформаційної системи, виконання виробничих робіт.

На ринку інформаційних послуг **аутсорсинг** – це така методологія роботи, коли підприємство всі турботи відносно програмно-технічного

комплексу покладає на зовнішню організацію, а саме лише користується інформацією цього комплексу. За договором і завданням підприємства зовнішня організація усуває будь-які неполадки і за необхідності проводить модернізацію. Аутсорсинг передбачає отримання потоків інформації з боку зовнішніх організацій, а не самостійну побудову і ведення програмно-технічного комплексу, буквально – це отримання *потоків інформації від програмно-технічного комплексу, що належить сторонній організації*.

Реалізація рішень, що пропонуються ринком програмних проектів, здійснюється з розподілом функцій взаємодії з програмно-апаратним комплексом ІС між користувачем, проблемним програмістом, системним програмістом та системним інтегратором (рис. 75).

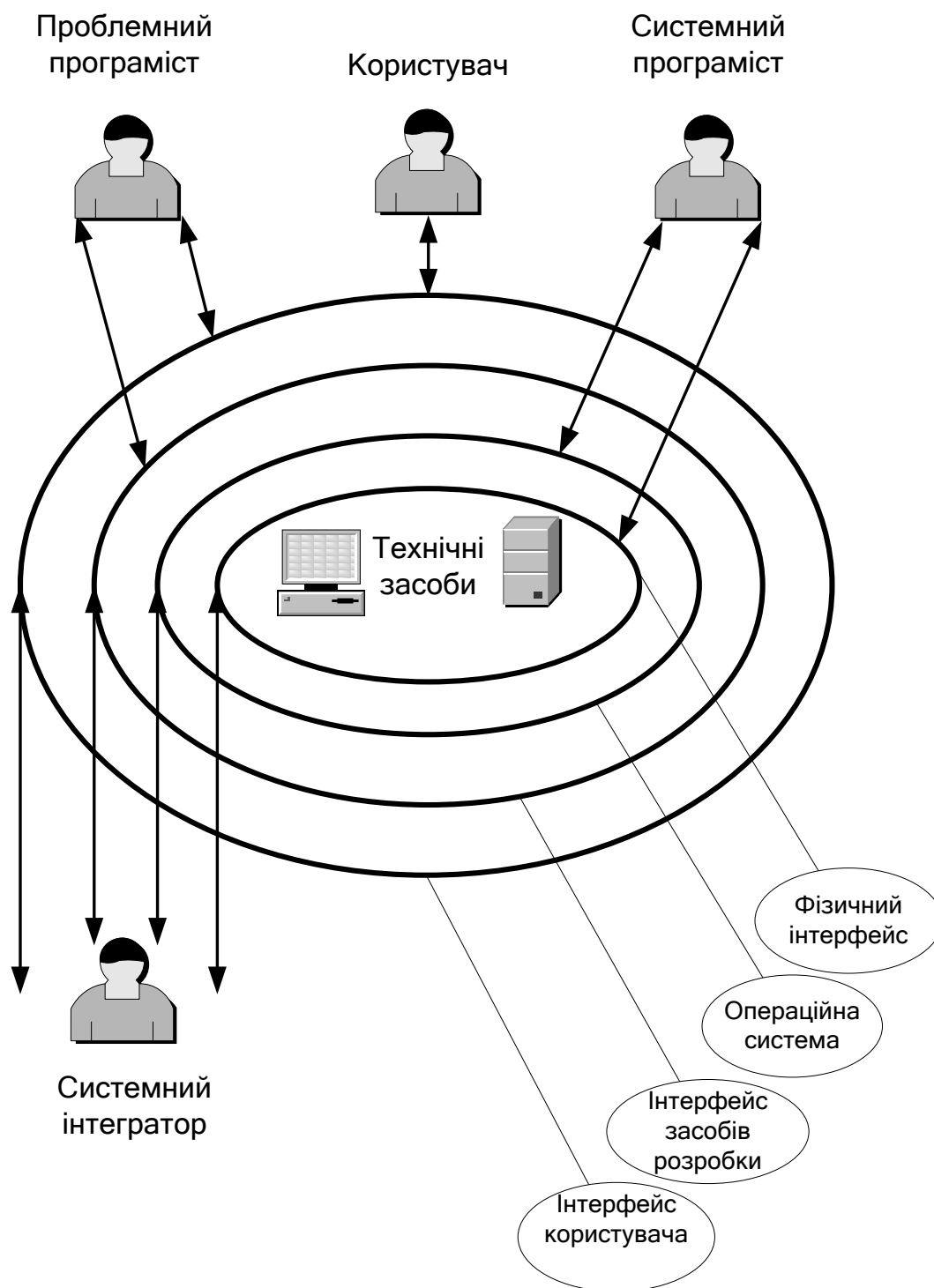


Рис. 75. Розподіл функцій взаємодії фахівців із програмно-технічним комплексом ІС

Функції користувача, проблемного програміста, системного програміста та системного інтегратора розподілені наступним чином.

Користувач програмно-технічного комплексу формулює проблеми і завдання бізнесу. Далі він самостійно або за допомогою **проблемного програміста** вибирає пакети прикладних програм (ППП) для вирішення цих завдань.

Системний програміст – це фахівець, який розуміє особливості побудови технічних засобів комп'ютера, проблеми передачі даних в мережах, знає операційні системи і мови програмування. Системний програміст обирає придатні операційні системи, виконує об'єднання програмно-технічних засобів в єдину систему, готує цю систему для завантаження до неї вхідних даних і ППП. Під час створення системи системний програміст співробітничав з системним інтегратором.

Системний інтегратор консультує і супроводжує пошуки користувача при вивченні можливостей ринку інформаційних систем. Він оцінює складність завдань користувача, рекомендує технічне і програмне забезпечення для придбання, допомагає або сам поєднує технічні модулі між собою, обирає та інсталує потрібні програми, навчає користувача і вирішує разом з ним комплекс реальних завдань на конкретних даних, настраює прикладні програми на реальні завдання користувача.

Участь системного інтегратора на всіх етапах процесу проектування дає можливість створювати більш ефективні інформаційні системи. Так, на самому початку проекту вони створюють консультаційну групу для проведення передпроектних досліджень. Тісне співробітництво з виробниками допомагає їм пропонувати технічні рішення на базі технологій і устаткування, що з'являться на ринку через рік чи два, тобто пропонуються найбільш сучасні рішення, які морально не застаріють до того моменту, коли буде спроектована і запущена ІС.

У процесі реалізації інтеграційних проектів фірми-інтегратори проводять наступні види робіт:

- проектують топологію інтегрованих обчислювальних систем;
- вибирають ефективні технічні рішення;
- визначають оптимальний склад апаратних і програмних засобів;
- здійснюють монтаж, супровід і гарантійне обслуговування техніки протягом усього терміну експлуатації системи;

- здійснюють постачання готових компонентів інформаційної системи, включаючи обчислювальну і мережну техніку, різні програмні продукти (мережні засоби, системи офісної автоматизації, банківські системи і прикладне програмне забезпечення);

- розробляють власне програмне забезпечення на базі сучасних CASE-технологій.

14.10.2. Проектна інтеграція

Іншим варіантом організації системної інтеграції є **проектна інтеграція**, яка становить реалізацію проектів від консалтингу до створення прикладної системи – зі здачею замовнику всієї інформаційної системи "під ключ" із залученням партнерів для реалізації певних складових проекту. Проектна інтеграція – це діяльність, що включає:

пошук складових частин для вирішення комплексного завдання;
розподіл відповідальності і складання плану-графіка робіт вирішення завдання.

Проектна інтеграція – це інтеграція наявних проектів, залучення і використання потрібних ресурсів.

Проектний інтегратор відрізняється від системного інтегратора тим, що, по-перше, максимально активно використовує аутсорсинг, по-друге, робить це максимально ефективно і з мінімальними витратами, так, щоб проект почав працювати в реальному часі і якнайшвидше дав економічний ефект. *Якщо системний інтегратор створює нові інформаційні системи, то проектний удосконалює роботу ІС шляхом пошуку на ринку вже наявних, упроваджених рішень і об'єднання їх.*

Щоб можна було зняти яку-небудь окрему проблему, не залучаючи засобів на передпроектне обстеження, проектний інтегратор спирається на допомогу співробітників відділу автоматизації замовника для виявлення проблеми, пошуку і вибору потрібних рішень, після чого проектний інтегратор зав'язується з тими, хто впровадив таке рішення, і розробляє техніко-економічне обґрунтування цього рішення. **Результатом проектної інтеграції** є певний спільний продукт, у який повинні входити розробки фірми – проектного інтегратора, розробки субпідрядника з урахуванням побажань відділу автоматизації, виконані без витрат на передпроектне обстеження, і який починає працювати в стислий термін.

З огляду на це можна виділити **основні функції**, які виконують фірми – проектні інтегратори:

1. Співпраця з відділами автоматизації підприємства-замовників, розгляд їх як партнерів, які самі для себе виконують велику роботу з автоматизації різних типових галузей, замовляти яку для великих структур було б не вигідно.

2. Робота з великими організаціями, сприймання їх відділів автоматизації як фірм, що випускають певну продукцію в цій галузі і

намагаються домовитися з ними про використання їхнього продукту і робочого часу з метою здійснення проекту.

3. Налагодження структурних зв'язків між різними розробниками для того, щоб вони могли на комерційній основі обмінюватися проектними рішеннями. Коли ж з'являється багато субпідрядників, то з'являється необхідність в управлінні проектом, спостереженні за виконанням, щоб усі частини були правильно сполучені, здані в термін і т. д.

14.11. Планування і контроль проектних робіт

14.11.1. Методи планування і управління проектами і ресурсами

В управлінні проектами існує декілька способів формалізованого подання виконуваної сукупності робіт, використовуваних з метою планування і управління. Значного поширення у процесі побудови моделей систем управління комплексом операцій отримали графічні методи як найбільш універсальні і такі, що дають інформацію про хід робіт, основними з яких є метод побудови лінійного графіка (діаграми) Гантта і метод мережного планування та управління (МПУ).

Діаграма Гантта, або циклограма – горизонтальна лінійна діаграма, на якій роботи проекту зображуються тривалими за часом відрізками, датами, що характеризуються, початком і закінченням, затримками і, можливо, іншими часовими параметрами (рис. 76).

В основі системи МПУ лежить побудова мережної моделі плану – мережного графіка, на якому в певному порядку наочно показані всі операції щодо створення спочатку проміжних результатів проектування з певним ступенем готовності і під кінець – повне завершення розробки. При побудові мережного графіка необхідно для кожної роботи (операції) знати роботи (операції), які безпосередньо передують даній і що йдуть за нею. Мережа виражає, таким чином, співвідношення порядку, що існує на чисельності робіт, які характеризуються часом виконання, і подій, які характеризуються часом початку і часом закінчення.

Код роботи	Початок роботи	Кінець роботи	Тривалість, дні	Часові періоди, дні							
				1	2	3	4	5	6	7	
1	1.12	2.12	1	█							
2	2.12	4.12	2		█						
3	4.12	7.12	3			█					
4	5.12	7.12	2				█				
5	6.12	7.12	1					█			

Рис. 76. **Діаграма Гантта**

На мережних графіках події зображуються кружечками з порядковими номерами, дійсні роботи, очікування – суцільними стрілками, фіктивні роботи або залежності – пунктирними лініями-стрілками. Стрілки вказують послідовність виконання операцій (рис. 77).

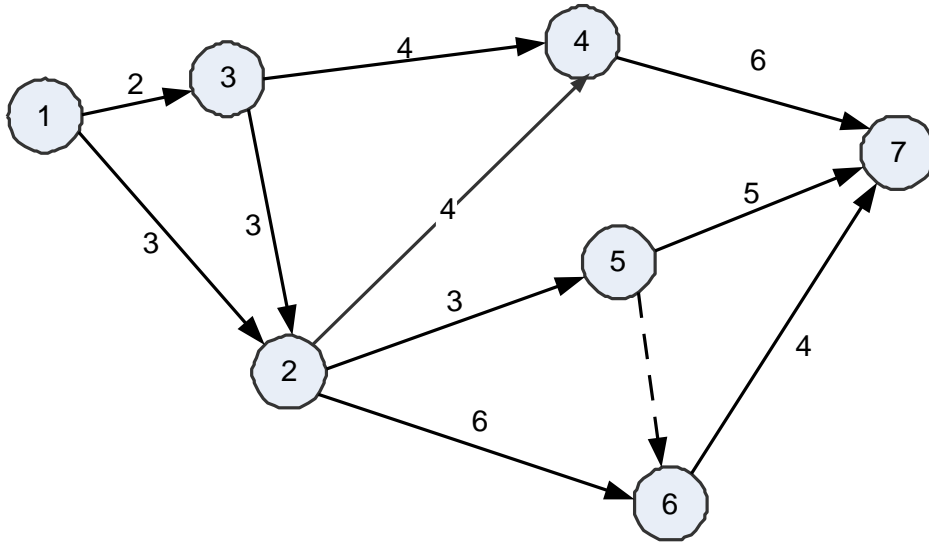


Рис. 77. **Мережний графік**

Взаємозв'язок кружечків і стрілок, що є графічними символами мережної моделі, повинен будуватися за певними правилами.

1. Будь-яка робота позначається стрілкою, яка з'єднує тільки дві події і відбиває процес переходу від однієї події до іншої.

2. Подія, з якої стрілка виходить, називається початковою або попередньою по відношенню до подальшої роботи. Подія, в яку стрілки входять, є кінцевою або наступною.

3. Початок стрілки показує, з якої події дана робота починається, а кінець стрілки – в якій події вона закінчується.

4. Роботи мають часові оцінки, які проставляються на стрілках. Подія вважається такою, що відбулася, тоді, коли буде закінчена найбільш тривала зі всіх робіт, що в неї входять.

5. Потрібні для виконання роботи розміри ресурсів вказуються на стрілках в дужках.

Слід розглянути ключові визначення і концепції, використовувані в методах планування, організації та контролю проекту.

Робота – це певна діяльність, необхідна для досягнення конкретних результатів. Таким чином, робота є основним елементом діяльності на найнижчому рівні деталізації проекту, на виконання якого потрібні час і ресурси і який може затримати початок виконання інших

робіт. Момент закінчення роботи означає факт отримання кінцевого продукту (результату роботи).

У поняття "робота" входить також **очікування**, тобто пасивний процес, що не вимагає витрат праці і матеріальних ресурсів, але який потребує часу.

Під роботою мають на увазі і **просту залежність**, тобто логічний зв'язок між двома або більшою кількістю операцій, яку іноді називають **холостою**, або **фіктивною роботою**, тому що вона не вимагає жодних витрат часу, вартості, праці.

Подія (віха) – результат виконання однієї або декількох робіт або дата в ході здійснення проекту. Подія використовується для відображення стану завершеності тих або інших робіт. У контексті проекту менеджери використовують події або віхи для того, щоб зазначити важливі проміжні результати, які повинні бути досягнуті в процесі реалізації проекту. Важливою відмінністю подій від робіт є те, що вони не мають тривалості.

Зв'язки передування (логічні залежності) відображають природу залежностей між роботами. Більшість зв'язків у проектах відносяться до типу "кінець – початок", коли наступна робота може початися тільки після закінчення попередньої роботи. Крім того, можуть бути зв'язки типу "кінець – кінець", "початок – початок", "початок – кінець". Зв'язки передування утворюють структуру мережі. Комплекс взаємозв'язків між роботами часто називають логічною структурою проекту, оскільки він визначає послідовність виконання робіт.

Максимальний за тривалістю повний шлях в мережі називається **критичним**; роботи, що лежать на цьому шляху, також називаються **критичними** (на графіку вони відображаються подвійними стрілками). Виявлення критичного шляху дозволяє встановити роботи (операції), що визначають хід виконання проекту. Критичні роботи в ході проектування повинні виконуватися чітко за графіком. Саме тривалість критичного шляху визначає найменшу загальну тривалість робіт за проектом у цілому.

Метод критичного шляху дозволяє розрахувати можливі календарні графіки виконання комплексу робіт на основі описаної логічної структури мережі і оцінок тривалості виконання кожної роботи, визначити критичний шлях проекту. Тривалість виконання всього проекту в цілому може бути скорочена за рахунок скорочення тривалості робіт,

що лежать на критичному шляху. Відповідно будь-яка затримка виконання робіт критичного шляху спричинить збільшення тривалості проекту. Концепція критичного шляху забезпечує концентрацію уваги менеджера на критичних роботах.

Усі критичні роботи є потенційно "вузькими" місцями плану. Критичних шляхів може бути декілька. Шляхи, тривалість яких наближається до критичного шляху, називаються **субкритичними**. Решта шляхів – некритичні. Наявність критичного шляху дозволяє використовувати його як основу для оптимізації плану. Роботи, які лежать на некритичному шляху, мають певні резерви часу, які є важливими показниками роботи мережі. Проте основною перевагою методу критичного шляху є можливість маніпулювання термінами виконання робіт, які не лежать на критичному шляху.

Часовий резерв, або запас часу, – це різниця між найбільш раннім можливим терміном завершення роботи і найпізнішим допустимим часом її виконання. Управлінський сенс часового резерву полягає в тому, що за необхідності врегулювати технологічні, ресурсні або фінансові обмеження проекту він дозволяє менеджеру затримати роботу на цей час без впливу на загальну тривалість проекту і тривалість безпосередньо пов'язаних із ним робіт.

Ресурси – забезпечувальні компоненти діяльності, що включають виконавців, енергію, матеріали, машинний час, устаткування і т. д. Відповідно з кожною роботою можна зав'язати функцію потреби в ресурсах.

Методики призначення і вирівнювання ресурсів дозволяють менеджеру проаналізувати мережний план, побудований за допомогою методу критичного шляху, з метою забезпечення доступності і використання певних ресурсів протягом усього часу виконання проекту.

Призначення ресурсів полягає у визначенні потреби кожної роботи в різних типах ресурсів. Вирівнювання **ресурсів** здійснюється, як правило, за допомогою програмно реалізованих евристичних алгоритмів планування за обмежених ресурсів. Ці засоби допомагають менеджеру створити реальний розклад проекту з урахуванням потреби проекту в ресурсах і фактично доступних в даний момент часу ресурсів.

Ресурсна гістограма – гістограма, що відображає потреби проекту в тому або іншому вигляді ресурсів у кожен момент часу.

Ресурсне календарне планування – планування термінів початку робіт за обмежених наявних ресурсів. Перевірка ресурсної реалізованості календарного плану вимагає зіставлення функцій

наявності і потреби в ресурсах для проекту в цілому. Зсовуючи некритичні роботи аж до їх пізніх термінів початку (закінчення), можна видозмінити ресурсний профіль, забезпечуючи оптимальне використання ресурсів. Інформація, отримана в результаті ресурсного аналізу проекту, допомагає загострити увагу менеджера і членів команди на тих моментах роботи, де ефективно управління ресурсами буде ключовим фактором успіху.

Аналіз здійсненності проекту може виконуватися в наступних напрямках:

логічна здійсненність (урахування логічних обмежень на можливий порядок виконання робіт у часі);

часовий аналіз (розрахунок і аналіз тимчасових характеристик робіт: рання/пізня дата початку/закінчення роботи, повний, вільний часовий резерв та ін.);

фізична (ресурсна) здійсненність (урахування обмеженості наявних або доступних ресурсів у кожний момент часу виконання проекту);

фінансова здійсненність (забезпечення позитивного балансу грошових коштів як особливого виду ресурсу).

Параметрами, за якими повинне здійснюватися управління проектом, є наступні: час, вартість, ресурси, техніко-економічні показники (ТЕП). Час управління проектом враховується завжди, решта параметрів – у необхідних випадках. Залежно від різного поєднання параметрів виникають і відповідні різновиди системи управління комплексом робіт за кінцевих її параметрами і їх поєднанням, до яких відносяться:

час;

час – вартість;

час – ресурси;

час – вартість – ресурси;

час – ТЕП;

час – вартість – ТЕП;

час – ресурси – ТЕП;

час – вартість – ресурси – ТЕП.

Найчастіше використовуються системи з параметром "час". Можливість урахування в системі всіх видів ресурсів, до яких насамперед слід віднести робочу силу, устаткування і грошові ресурси, значно розширює сферу планування та управління системою.

14.11.2. Технологія застосування методу МПУ для розробки проекту ІС

В управлінні проектами рекомендовані наступні етапи планування робіт:

1) задавання основних робіт, їх деталізація і визначення послідовності виконання робіт;

2) задавання приблизного часу виконання кожної роботи і оцінка підсумкового часу виконання всього проекту;

3) визначення груп персоналу (керівник, програмісти, оператори та ін.), що беруть участь у даному проекті;

4) зазначення для кожної роботи персоналу, який повинен її виконати. Розрахунок людино-дня для кожної категорії працівників і оцінка чисельності необхідного персоналу;

5) зазначення потрібних для кожної роботи матеріальних ресурсів. Розрахунок підсумкових вимог матеріальних ресурсів і визначення необхідної кількості матеріальних ресурсів кожного виду, ефективності їх використання;

б) оцінка фінансових витрат за наступними статтями витрат:

а) витрати на зовнішні послуги, роботи, наприклад, на купівлю юридичних, адресних та інших баз даних;

б) витрати на технічні й програмні засоби, а також на канали зв'язку;

в) витрати на персонал. Для цього потрібно вказати вартість одного людино-дня для кожної категорії працівників, що беруть участь у даному проекті;

г) інші витрати;

д) витрати разом. Можна підсумовувати всі ці статті витрат і оцінити витрати на розробку ІС;

7) оптимізація мережного графіка робіт на основі аналізу критичного шляху. Оптимізація робіт має на увазі наступні моменти:

розгалуження;

рівномірне завантаження наявного персоналу;

рівномірне завантаження технічних засобів;

аналіз необхідних резервів.

Крім основних робіт, в початковий план проектування повинні бути включені **процедури контролю** проектної діяльності, які відіграють важливу роль у системі управління проектуванням. Процедури контролю необхідно планувати як за змістом, так і за часом.

Календарний план проведення контролю є переліком моментів часу контрольних точок, у яких доцільно здійснювати контроль за станом проекту. Даний план повинен доповнюватися методичними матеріалами за змістом контрольних процедур в контрольних точках.

В основі методики проведення контролю лежать результати аналізу складу і змісту технологічних операцій проектування, які повинні бути виконані до контрольної точки. Тому мережний графік повинен містити контрольні моменти у вигляді запланованих термінів подання результатів проектування або впровадження робіт і бути зручним для систематичного вивіряння і потрібного коректування. Після цього мережний графік розглядається і затверджується керівництвом проектування і підприємства, для якого ці роботи проводяться.

14.11.3. Розрахунок основних показників мережного графіка.

До основних показників розрахунку мережного графіка відносяться наступні:

- тривалість кожної роботи;
- ранній час закінчення події;
- пізній час закінчення події;
- час раннього початку роботи;
- час пізнього початку роботи;
- час раннього закінчення роботи;
- час пізнього закінчення роботи $t_{i\phi(j)}$;
- повний резерв часу;
- частковий резерв часу роботи;
- частковий резерв часу події.

Кожна робота $i-j$ мережного графіка має часову оцінку – тривалість $t_{(i-j)}$, що виражається в одиницях часу (днях, тижнях, декадах, місяцях). При цьому тривалість виконання кожної роботи відома заздалегідь або може бути визначена.

Будь-яка послідовність робіт в мережному графіку, в якому кінцева подія попередньої роботи співпадає з початковою подією наступної, називається **шляхом**. Тривалість (довжина) будь-якого шляху L дорівнює сумі тривалості робіт його складових:

$$T L = \sum t_{(i-j)} .$$

У мережному графіку виходить декілька шляхів від початкової події до кінцевої. **Критичний шлях** – це такий шлях, сумарна тривалість якого має максимальне значення:

$$T_{кр} = \max T L ,$$

де $T_{кр}$ – тривалість критичного шляху.

Критичний шлях визначає час, необхідний для виконання всіх проектних робіт, включених у графік. Усі роботи, що лежать на цьому шляху, є **критичними**, і від їх тривалості залежатиме найвигідніший термін виконання програми проектування. Скорочення або збільшення тривалості критичних робіт відповідно скоротить або збільшить загальну тривалість виконання програми.

У часових оцінках, як правило, закладається резерв, який може компенсувати окремі неточності попереднього планування і дозволяє уникнути відхилень фактичного виконання робіт від запланованого за графіком.

Ранній і пізній терміни закінчення конкретної події визначаються за максимальним зі шляхів, що проходить через подію. Ранній термін дорівнює тривалості максимального із шляхів, які передують події, а пізній термін – різниці між $T_{кр}$ і тривалістю максимального з наступних за дією шляхів, тобто:

$$t_{pi} = \max \left\{ T \left[L_{1i} \right] \right\},$$

де t_{pi} – ранній час здійснення події i ;

L_{1i} – шлях, що передує події i ;

$$t_{ni} = T_{кр} - \max \left\{ T \left[L_{2i} \right] \right\},$$

де t_{ni} – пізній час звершення події i ;

L_{2i} – наступний шлях за подією i .

Для критичного шляху $t_{pi} = t_{ni}$.

Ранній термін початку $t_{рп\ i-j}$ роботи визначається тривалістю максимального шляху від початкової події мережі до початкової події для даної роботи:

$$t_{рп\ i-j} = t_{р\ i} ,$$

де $t_{рп\ i-j}$ – час раннього початку роботи $i-j$;

$t_{р\ i}$ – ранній час звершення події i .

Ранній термін закінчення $t_{рз\ i-j}$ роботи дорівнює сумі раннього терміну початку роботи і тривалості даної роботи:

$$t_{рз\ i-j} = t_{р\ i} + t_{i-j}$$

де $t_{рз\ i,j}$ – час раннього закінчення роботи $i-j$;

t_{i-j} – тривалість роботи $i-j$.

Пізній термін початку $t_{пп\ i-j}$ роботи, за якого критичний шлях не зміниться, може бути визначений як різниця між пізнім закінченням даної роботи і її тривалістю:

$$t_{пп\ i-j} = t_{пз\ i-j} - t_{i-j} ,$$

де $t_{пп\ i-j}$ – час пізнього початку роботи $i-j$;

$t_{пз\ i-j}$ – час пізнього закінчення роботи $i-j$;

t_{i-j} – тривалість роботи $i-j$.

Пізній термін закінчення роботи, за якого тривалість критичного шляху не зміниться, дорівнює пізньому терміну настання кінцевої події даної роботи:

$$t_{пз\ i-j} = t_{п\ j} ,$$

де $t_{пз\ i-j}$ – час пізнього закінчення роботи $i-j$;

$t_{п\ j}$ – пізній час звершення події j .

Для критичного шляху характерні наступні співвідношення:

$$t_{рп\ i-j} = t_{пп\ i-j} ;$$

$$t_{рз\ i-j} = t_{пз\ i-j} .$$

14.11.4. Оптимізація мережного графіка

Оптимізацією називають процес коректування початкового мережного графіка шляхом послідовного поліпшення мережі з метою досягнення заданого терміну або рівномірного розподілу різних видів ресурсів. Завданням оптимізації є забезпечення відповідності між установленими термінами робіт і відпущеними для їх виконання ресурсами.

Як правило, оптимізація здійснюється за наступними критеріями:

часом ($T_{кр} \leq T_{дир}$, тобто час критично менший або дорівнює часу директивному);

витратами матеріальних ресурсів;

витратами грошових ресурсів;

техніко-економічними показниками.

1. Оптимізація за часом.

Спочатку мережа коректується за критерієм "час" без урахування обмежень. Існує декілька шляхів вдосконалення мережних графіків. Один із них заснований на оцінці величини директивних термінів закінчення робіт $t_{дир}$.

Якщо $t_{кр} <$ або $= t_{дир}$, виникає додатковий резерв часу, який розраховується за формулою:

$$R_{\text{доп}} = t_{\text{дир}} - t_{\text{кр}}.$$

Додатковий резерв може бути використаний для збільшення тривалості окремих критичних робіт за подальшої оптимізації.

Якщо $t_{\text{кр}} > t_{\text{дир}}$, то переглядається мережа з метою її ущільнення. Головне завдання – прискорення тих робіт, з яких у кожному випадку складається критичний шлях.

Ущільнення мережного графіка або його перепланування проводиться звичайно кілька разів методом послідовних наближень, тобто багаторазовим стисненням чергового критичного шляху, доки не буде досягнутий задовільний результат.

Існує декілька методів узгодження мережного графіка із заданими термінами:

1) зміна часових оцінок шляхом заміни нормативної тривалості t_{i-j}^H скороченою тривалістю робіт – t_{i-j}^C ;

2) зміна топології мережі внаслідок перегляду виконання робіт;

3) розчленовування робіт і їх поєднання за часом.

Загальний термін виконання програми слід скорочувати за рахунок зміни тривалості критичних робіт, оскільки він не пов'язаний зі зміною топології мережі. Рекомендується зменшувати тривалість не тільки критичних робіт, але й у тих, що лежать на підкритичних шляхах, оскільки вони можуть стати критичними. Зменшення часових оцінок відбувається за рахунок перекидання відповідних ресурсів з ненапружених робіт, що характеризуються значними резервами часу.

Якщо не вдається повною мірою зменшити термін виконання розробки за рахунок форсування робіт, то вдаються до зміни топології мережі. Це можливо тому, що окремі роботи можуть виконуватися різними методами. Багатоваріантна технологія дозволяє відшукувати нову послідовність виробництва робіт і нові взаємозв'язки.

Найчастіше проводять оптимізацію мережі на основі розрахунку резервів часу для робіт, що знаходяться на некритичному шляху.

У процесі визначення резервів часу враховують, що повний резерв часу роботи R_{i-j} – термін, на який можна пересунути дану роботу, не збільшуючи, визначається за формулою:

$$R_{i-j} = t_{пз\ i-j} - t_{рз\ i-j} = t_{п\ j} - t_{р\ j} - t_{i-j}.$$

Окремий резерв часу роботи r_{i-j} – термін, на який можна пересунути дану роботу, не впливаючи на інші характеристики мережі, виражається залежністю:

$$r_{ij} = t_{р\ j} - t_{рз\ i-j} = t_{р\ j} - t_{ij}.$$

Резерв часу події r_i – термін, на який можна зсунути звершення даної події, не збільшуючи тривалості всієї розробки, складе:

$$r_i = t^п_i - t^р_i.$$

Наступним методом за наявності необхідних трудових ресурсів є **розгалуження робіт** критичного шляху.

При скороченні терміну за рахунок тих або інших заходів і виявленні декількох варіантів в мережного графіка забезпечується виконання робіт у заданий термін. Необхідно зарівняти ці варіанти і вибрати кращий за допомогою комп'ютера.

Одночасно зі скороченням критичного шляху зменшуються і резерви часу, внаслідок чого поступово виникає все більше і більше критичних робіт і шляхів. Тому необхідно перевірити тривалість шляхів, що залишилися, і оцінити ступінь напруженості некритичних груп робіт. Визначити ступінь напруженості виконання кожної некритичної групи робіт можна за допомогою коефіцієнта напруженості робіт $K_{н\ i-j}$.

Коефіцієнт напруженості визначається відношенням відрізків, які не співпадають із критичним шляхом максимального шляху, що проходить через дану роботу, до критичного шляху:

$$K_{н\ i-j} = \frac{t L_{\max} + t^1 L_{кр}}{t L_{кр} - t^1 L_{кр}},$$

де $t^1 L_{кр}$ – величина відрізка, що співпадає з критичним шляхом;

$t L_{\max}$ – протяжність максимального шляху, що проходить через дану роботу.

Якщо після всіх вжитих заходів щодо скорочення тривалості виконання програми директивний термін не досягнутий, ставиться питання перед керівництвом про зміну цього терміну.

2. Оптимізація за ресурсами.

Складений план розробки повинен бути оптимізований не тільки за термінами, але й за всіма видами ресурсів. Часто виявляється, що оптимальні за часом варіанти розробок є на практиці важко реалізованими, тому що вони не враховують обмежень трудових ресурсів, тобто, наприклад, кількість проектувальників, які можуть бути зайняті обстеженням і проектуванням. Тому після розрахунку часових параметрів укрупненого мережного графіка проводиться його аналіз з метою встановлення відповідності параметрів мережного графіка заданим обмеженням використання якого-небудь ресурсу.

Основним способом оптимізації плану при врахуванні обмежень на ресурси служать **зняття ресурсів і перекидання** їх на критичний шлях з метою скорочення часу на критичному шляху. При цьому тривалість некритичних робіт збільшується, а критичних – зменшується. У результаті перерозподілу можуть з'явитися нові критичні шляхи. Тому процедура перерозподілу ресурсів повторюється доти, поки терміни проектування не виявляться рівними директивним або будуть меншими за них.

Потреба в ресурсах за мережним графіком визначається шляхом зображення мережного графіка в масштабі часу за ранніми або пізніми термінами початку й закінчення робіт, тобто шляхом побудови календарного лінійного плану. У календарному плані вказуються коди робіт, тривалості робіт, загальні резерви часу, а також наводиться циклограма, що відображає в стрічковому вигляді час початку, закінчення, тривалості робіт і величину їх резерву.

Крім того, щоб здійснювати перекидання ресурсів, складають під календарним планом гістограму потреби в ресурсах, в якій по осі X відображаються часові відрізки виконання робіт, а по осі Y зображуються сумарні потреби в трудових ресурсах на всіх відрізках часу біля лінії обмеження їх використання. Розподіл обмежених ресурсів з постійною інтенсивністю споживання, тобто кількістю ресурсу, використовуюваного в одиницю часу в даній роботі, зводиться до знаходження раціонального розподілу його за рахунок зниження пікової сумарної інтенсивності споживання до заданого рівня.

При правильному вирішенні проблеми часто вдається узгодити споживання ресурсу (кількість проектувальників, одночасно зайнятих розробкою проекту або обстеженням) із заданим обмеженням без збільшення три-

валості критичного шляху. Це досягається впорядкуванням чисельності людей за рахунок використання резервів часу некритичних робіт.

14.11.5. Інструментальні засоби управління проектами

Важливою складовою частиною системи управління проектами є інструментальні засоби, за допомогою яких реалізуються методи мережного планування та управління, а також календарного планування, що є сукупністю програмних засобів, спрямованих на підтримку й підвищення ефективності процесів планування та управління проектом. Вибір типів програмного забезпечення з управління проектами в організації здійснюють в наступній послідовності:

- аналіз вимог користувачів;
- аналіз ринку;
- вибір програмного забезпечення.

У системі управління проектами можна виділити три рівні управління проектами, відповідних певним категоріям користувачів ПЗ, таких, що виконують специфічні функції.

1. Рівень вищого керівництва, на якому відбувається визначення цілей і завдань підприємства, ухвалюється рішення про фінансування, оцінюється пріоритетність проектів.

2. Стратегічний рівень, що складається з професіоналів з управління проектами, що займаються плануванням і контролем корпоративних проектів. Як правило, цей рівень подається невеликою кількістю людей, основний обов'язок яких – саме управління проектами і які в своїй роботі спираються на програмне забезпечення з управління проектами. Роль подібних професіоналів є ключовою в організації, і вони працюють як група підтримки з управління проектами.

3. Рівень операцій, для якого робота з програмним забезпеченням щодо управління проектами вторинна. Це відповідальні за проекти на місцях, менеджери проектів, керівники груп. На рівні операцій потрібен інструмент з управління і контролю за проектом, але на невеликі відрізки часу.

У табл. 29 наведені вимоги до програмних засобів планування та управління проектними роботами з боку фахівців трьох вище перелічених рівнів.

До основних чинників, що зумовлюють вибирання інструментального засобу для управління проектами, можна віднести наступні:

- тип завдань, для яких буде потрібна система управління проектами;
- характер діяльності організації з погляду можливості і доцільності застосування проектної форми планування і управління;
- вид діяльності, яка може плануватися у вигляді проектів;
- рівень детальності, до якого необхідно планувати і контролювати проекти.

Вимоги до програмних засобів планування та управління проектними роботами

1. Рівень вищого керівництва	2. Стратегічний рівень	3. Операційний рівень
1.1. Легкість у застосуванні. 1.2. Можливість отримувати демонстраційні звіти. 1.3. Можливості узагальнення відомостей 1.4. Засоби для інтеграції з даними з інших програмних продуктів. 1.5. Процедури для планування зверху-вниз	2.1. Засоби часового, ресурсного, вартісного планування, аналізу ризиків. 2.2. Можливість інтеграції з іншими програмними продуктами. 2.3. Засоби для згортання даних за проектом (надання звітів керівництву) і заглибленням для планування на детальнішому рівні. 2.4. Засоби для контролю за реалізацією проекту. 2.5. Гнучкість при налаштуванні вихідних форм звітності	3.1. Простота використання. 3.2. Легкість вивчення. 3.3. "Прозорість" процедур введення даних. 3.4. Наочність

Для підтримки різних управлінських завдань використовуються різні програмні засоби.

1. Для укрупненого опису і аналізу проекту на передінвестиційній стадії більшою мірою підходить спеціалізоване ПЗ аналізу проектів, яке дозволяє виконати оцінки основних показників рентабельності проекту загалом і обґрунтувати ефективність капіталовкладень.

2. Якщо крім обґрунтування інвестицій, існує потреба в контролі за ходом реалізації проекту, то необхідно переходити до використання ПЗ управління проектами.

Якщо принципове рішення про використання системи для управління проектами ухвалене, то для вибору пакету необхідно відповісти на питання, пов'язані зі з'ясуванням складу функцій планування і управління, які потрібно реалізувати:

лише планування або планування і контроль за ходом проекту;

планування і контроль лише термінів виконання робіт;

планування і контроль фінансових вкладень без детального планування використання ресурсів;

детальне планування використання ресурсів;

багатопроектне управління.

Далі слід визначити також вимоги до наступних компонентів проекту:

до розмірності проектів і детальності планування;

організаційної структури управління і звітності;

скільки проектів вестиметься одночасно і чи будуть вони взаємозалежними;

яка орієнтовна кількість завдань в одному проекті;

скільки видів ресурсів буде задіяно в одному проекті;

як розділятимуться ресурси між проектами.

Крім того, на вибір пакета можуть вплинути специфічні вимоги управління в конкретній предметній області (наприклад, спеціальні вимоги до звітності або необхідність розрахунку додаткових показників, необхідність інтеграції системи з іншими програмними продуктами або нормативними базами даних і т. д.).

Суттєвими є також міркування, пов'язані з кваліфікацією персоналу, який використовуватиме ПЗ. Пакети, що мають великі можливості, вимагають, як правило, вищої кваліфікації користувачів і додаткового навчання. Вони орієнтовані на користувачів-професіоналів, тобто фахівців, основним видом діяльності яких є адміністрування проекту.

Для користувачів, що використовують пакети управління проектами лише час від часу, за необхідності спланувати невеликий комплекс робіт, важливішими є простота використання і швидкість отримання результату.

У великих організаціях, як правило, можна знайти обидва типи користувачів. Завдання для таких організацій полягає не в тому, щоб зупинитися на якому-небудь одному пакеті, а в тому, щоб підібрати оптимальну комбінацію пакетів, що дозволяють обмін даними.

Вибираний засіб управління проектами повинен включати такі базові функціональні можливості.

1. Засоби опису комплексу робіт проекту, зв'язків між роботами і їх часових характеристик.

1.1. Засоби опису і типи планування:

виконати роботу "якомога раніше";

виконати роботу "якомога пізніше";

роботи з фіксованою датою початку-закінчення;

можливість узгодження тривалості завдань з обсягом призначених ресурсів;

обчислювані резерви часу (повний, вільний) і т. д.

1.2. Засоби встановлення логічних зв'язків між завданнями.

1.3. Багаторівневе подання проекту.

1.4. Підтримка календаря проекту, підтримка календарів ресурсів.

2. Засоби підтримки інформації про ресурси та витрати за проектом і призначення ресурсів і витрат за окремими роботами над проектом.

2.1. Ведення списку наявних ресурсів, можливість задавання нормального і максимального обсягів ресурсу.

2.2. Підтримка ресурсів з фіксованою вартістю і ресурсів, вартість яких залежить від тривалості їх використання.

2.3. Розрахунок необхідних обсягів ресурсів.

2.4. Ресурсне планування (виділення переобтяжених ресурсів і завдань, що використовують їх, автоматичне/командне вирівнювання профілів завантаження ресурсів з урахуванням обмежень за часом або з урахуванням обмеження на ресурс, з урахуванням пріоритетів завдань).

3. Засоби контролю за ходом виконання проекту.

3.1. Засоби відстеження стану завдань проекту (фіксація плану розкладу проекту, засоби введення фактичних показників стану завдань (відсоток завершення)).

3.2. Засоби контролю за фактичним використанням ресурсів (бюджетна кількість і вартість ресурсу, фактична кількість і вартість ресурсу, кількість і вартість ресурсів, потрібних для завершення роботи).

4. Графічні засоби подання структури проекту, засоби створення різних звітів за проектом.

4.1. Діаграма Гантта (часто поєднана з електронною таблицею і така, що дозволяє відображати різну додаткову інформацію).

4.2. PERT-діаграма (мережна діаграма).

4.3. Засоби створення необхідних для планування звітів (звіту за станом виконання розкладу, звітів про ресурси і за призначенням ресурсів, профіль ресурсу, звіт за вартістю).

До світових лідерів на ринку систем управління проектами належать продукти Primavera, CA Clarity, HP PPM, Microsoft Project, Planview, Compuware.

Проведені експертами Gartner дослідження ринку програмних продуктів у сфері управління проектами виділили їх сильні і слабкі сторони (табл. 30). Gartner, Inc. (NYSE: IT) – дослідницька і консалтингова компанія, світовий лідер у галузі інформаційних технологій. Є постійним партнером 60 тис. клієнтів в 10 тис. різних організаціях. За допомогою ресурсів Gartner Research, Gartner Consulting і Gartner Events компанія працює з кожним клієнтом над дослідженням, аналізом і впровадженням ІТ в бізнес. Компанія Gartner заснована в 1979 р., знаходиться в Стамфорді, Коннектикуті, США, і має 3,9 тис. партнерів, зокрема 1200 дослідників-аналітиків і консультантів у 75 країнах.

Слід розглянути детальніше найбільш популярні програмні продукти в галузі управління проектами.

Програмний продукт **Primavera** призначений для автоматизації процесів управління проектами відповідно до вимог PMI, IPMA і

стандартами ISO. Дане рішення має модульну структуру, модулі, засновані на web-технологіях. Зберігання даних здійснюється в єдиному сховищі, побудованому на базі Oracle або Microsoft SQL Server. Для збирання фактичних даних і актуалізації графіків система пропонує декілька модулів (для збирання даних в режимі реального часу, в режимі відсутності постійного підключення до мережі і для користувачів кишенькових комп'ютерів). Існує система адміністративної підтримки, яка дозволяє мінімізувати ризики, пов'язані із супроводом великомасштабних проектів. Є функціональність для вирішення завдань календарно-мережного планування.

Продукт **CA Clarity** забезпечує основу для всебічного управління ІТ-ресурсами за рахунок єдиної системи стратегічного планування і фінансового контролю ІТ-послуг. Дане програмне забезпечення є невід'ємною частиною рішення щодо оптимізації бізнесу (Business Service Optimization – BSO). CA розглядає рішення щодо управління проектами як ядро для розробки інтегрованого набору додатків для управління корпоративними ІТ-ресурсами, а рішення Clarity є головним інструментом, що забезпечує не просто планування проектів, а планування і контроль ІТ-ресурсів на рівні керівників підприємств і підрозділів.

Аналіз програмних продуктів управління проектами

Продукт	Сильні сторони	Слабкі сторони
1	2	3
Primavera	Надає могутній набір рішень з управління проектами. ProSight додає оптимальний набір інструментарій для автономного аналізу портфелів і системи управління. Швидка й ефективна служба підтримки.	Сфокусований на плануванні складних корпоративних проектів і значно менше на управлінні ІТ-запитами. Сильно спирається на партнерські відносини. Висока ціна може виявитися недоцільною для організацій з обмеженим бюджетом.
CA Clarity	Нові можливості інтегрують види ІТ-компонент у послуги, щоб краще асигнувати і відстежувати витрати і давати депозитаріям можливість проглядати вартість і статус ІТ-сервісів. CA зберіг місце на ринку з темпом зростання приблизно в 20% порівняно з HP і Planview. CA бачить управління проектами як наріжний камінь інтегрованого набору застосувань для управління ІТ-підприємства з Clarity як основний засіб підтримки виконавчого рівня планування і контролю за виконанням.	Тривалі сумніви, засновані на низькій оцінці та звітності минулих років заважають просуванню продукту на ринку. Clarity орієнтується в основному на організації середнього рівня.
HP PPM	Придбання продукту вигідне тим, що HP може надавати значні знижки. В останній версії HP PPM представлено розширене управління ресурсами і планування проектів, раніше невідомі за своєю ефективністю в цих галузях.	Незважаючи на великі можливості, HP повинен ще продемонструвати довгострокові зобов'язання на ринку управління проектами.

1	2	3
MS Project	<p>Ціна за ліцензію Microsoft Project становить потенційно рентабельну альтернативу іншим вендорам, націленим на корпоративних клієнтів.</p> <p>Компанія використовує широкі партнерські відносини для збільшення каналу маркетингу і утримання провідного місця на ринку.</p> <p>Microsoft придбав кращі системи з управління і аналізу портфелів, перейменував їх в Portfolio Server і продовжує розширювати наявну інтеграцію між ним і Project Server.</p> <p>Microsoft продовжує розробляти прикладне рішення управління проектами, об'єднуючи Microsoft Enterprise Project Management і Visual Studio Team System.</p>	<p>PPM-система Microsoft здійснює постачання і обслуговування через партнерів, що не завжди є рентабельним.</p> <p>На відміну від інших вендорів, PPM рішення Microsoft засновані на використанні безлічі незалежних програм, які повинні бути відповідним чином настроєні для використання їх як компонент PPM.</p>
Planview	<p>База знань Planview, заснована на кращих рішеннях, шаблонах і технологічних моделях, пропонує клієнтам посилання на довідники і бібліотеки, які вони можуть використовувати для подальшої автоматизації процесів PPM, оскільки вони прийняті в організації.</p>	<p>Для користувачів продукту Business Engine Planview розроблятиме рентабельний перехід на Planview enterprise.</p> <p>Planview підтримує лише платформу Windows і не підтримує жодних web-серверів, крім IIS.</p>
Compuware	<p>Постійне заглиблення функціональних можливостей.</p> <p>Доступні різні моделі постачань продукту, що дозволяє користувачам витратити гроші лише на те, що їм необхідне.</p>	<p>Changepoint підтримує лише платформу Microsoft, web-сервер IIS і Microsoft SQL сервер.</p>

HP Project and Portfolio Management (PPM) – продукт компанії HP за управління проектами і портфелем проектів призначений для вирішення проблеми, з якою постійно стикаються ІТ-підрозділи, – неможливості виконати частину проектів у встановлені терміни, не виходячи за межі виділеного бюджету, з найбільш оптимальним використанням наявних ресурсів. В основі PPM лежить платформа Project and Portfolio Management Foundation, що забезпечує спільне використання інформації і автоматизацію робочих потоків з використанням кращих практик управління бізнес-процесами ІТ-служби, безпеки і підготовки звітів. Усе це забезпечує відповідність стандартам і вимогам таких програм контролю якості та управління процесами, як Six-Sigma, CMMI, ITIL, ISO-9000 і Cobit.

Продукт **Microsoft Project** дозволяє отримувати інформацію, управляти проектними роботами, планами і фінансами, а також зберігати узгодженість роботи колективу. Завдяки інтеграції з Microsoft Office рішення підвищує продуктивність. Для управління корпоративними проектами в Microsoft існує рішення Microsoft Office Enterprise Project Management (EPM), комплексне середовище управління спільними проектами і портфелями. Рішення Office EPM дозволяє розширити аналіз і контроль всіх виконуваних робіт завдяки оптимізації процесу ухвалення рішень, підвищенню ступеня відповідності розробок стратегії розвитку бізнесу, більш обґрунтованому використанню ресурсів. До складу EPM входять наступні продукти: Microsoft Office Project Professional (управління проектами), Microsoft Office Project Server (засіб для централізованого управління проектами), Microsoft Office Project Portfolio Server (управління портфелями проектів).

Продукт **Planview** пропонує інструменти прогнозування, планування і оптимізації управління портфелями проектів на базі єдиної корпоративної стратегії. Об'єднуючи в рамках єдиного web-інтегрованого рішення управління різними бізнес-процесами – стратегіями, послугами, проектами, Planview дозволяє за рахунок ефективного адміністрування бюджетів, ресурсів і можливостей досягати оптимальної керованості бізнесу в умовах постійно змінного середовища.

Продукт **Compuware Changepoint** становить комплексне рішення, яке забезпечує налаштування й управління ІТ і дозволяє збільшити ефективність роботи пакета ІТ в цілому. Changepoint також спрощує

управління захистом передачі інформації, пов'язаної зі сферою діяльності ІТ, починаючи із введення запиту служб, закінчуючи розстановкою пріоритетів і вибором портфелів для продовження моніторингу проекту і стану справності прикладних програм. Compuware Changerpoint автоматизує основний принцип операцій інформаційних технологій, який стосується організаційної структури і переміщує релевантні дані аж до візуалізації пакета послуг ІТ, яка дає можливість надавати інформацію обмеженій кількості громадян у потрібний час. Рішення дозволяє контролювати не лише проекти, але й позапроектну роботу.

Компанія Gartner представила "магічний" квадрант для ПЗ управління проектами і портфелями проектів (рис. 78). "Магічний" квадрант Gartner відомий як одна з найбільш комплексних методологій оцінки ринку рішень для ПЗ.

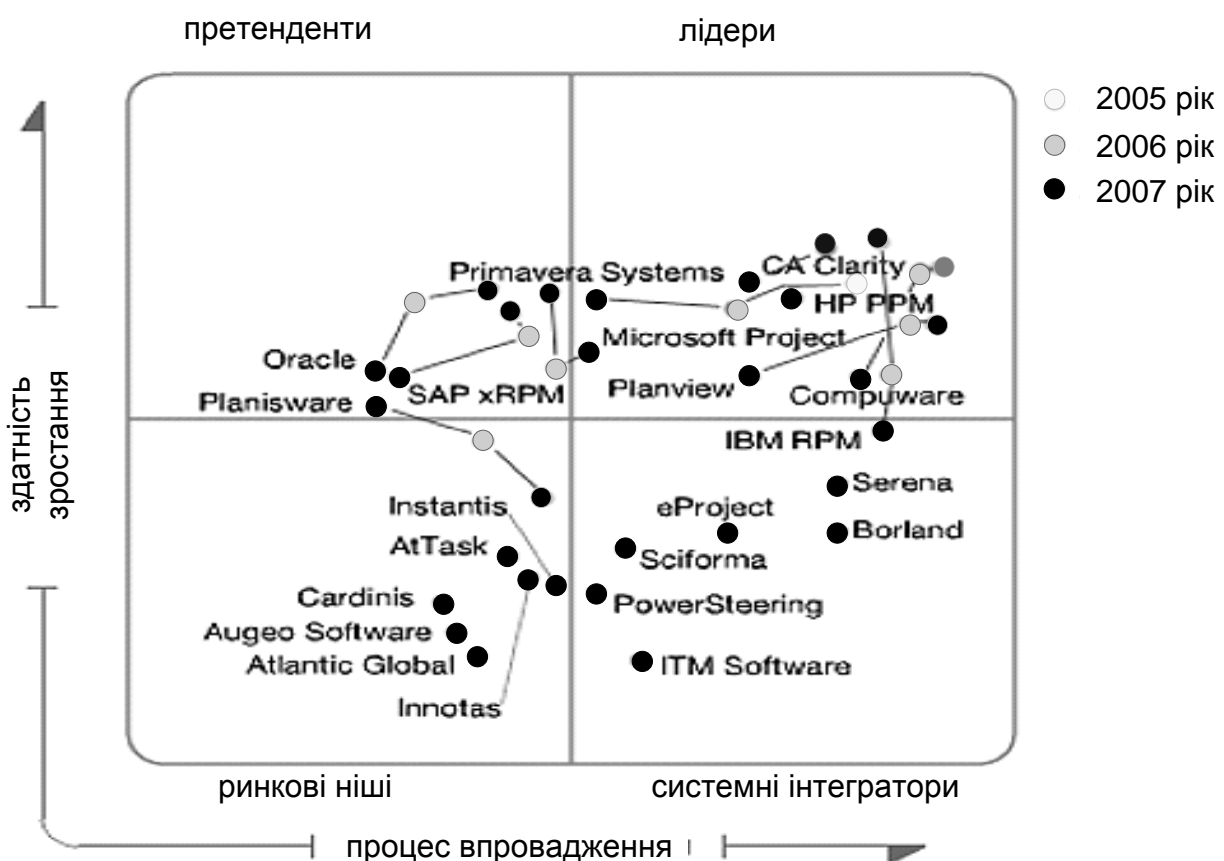


Рис. 78. ПЗ для управління проектами і портфелями проектів

У Gartner відзначають, що світовий ринок продажів систем розробки додатків та управління проектами в 2006 році досяг \$ 7,2 млрд., збільшившись на 7% порівняно з 2005 роком. Світовий ринок систем управління проектами за період 2005 – 2006 рр. виріс на 20%. Для порівняння – системи розробки додатків за цей же період показали зростання лише на 2%. Подібні темпи зростання вище середнього рівня очікуються аж до 2011 року. За прогнозами аналітиків Gartner, інтеграція функцій управління проектами, управління ІТ-сервісами і життєвим циклом додатків (ALM) в єдине рішення для планування та управління ІТ-ресурсами (ITPC) поки перебуває у стадії розробки, і сформується цей ринок приблизно до 2009 року.

За даними компанії Gartner, протягом останніх років провідним гравцем ринку ПЗ з управління проектами є Primavera, найбільший незалежний постачальник комплексних рішень для управління проектами, ресурсами і портфелями проектів. Позиція компанії Microsoft Project, на думку аналітиків, не є стабільною. Якщо у 2002 – 2003 рр. в звітах Gartner ця система посідала провідні місця, то в 2005 – 2006 роках вендор був відзначений як такий, що лише стійко посідає свою нішу. Аналітики також відзначили в рейтингах 2006 – 2007 рр. компанію CA (з продуктом Clarity), яка займається уніфікацією і спрощенням управління ІТ в організаціях.

Ринок програмних продуктів для управління проектами зростає і розвивається. Системи більшою мірою орієнтовано на Інтернет. Такі системи дозволяють забезпечити доступ до проектної документації для всіх членів команди в режимі реального часу. Очікується, що в майбутньому системи від різних виробників будуть все більше схожі один на одного. Це пояснюється тим, що базова основа всіх систем управління проектами і календарного планування одна й та ж. Крім того, користувачі хочуть бачити проектні дані в звичному і зрозумілому їм форматі.

Контрольні запитання

1. Що таке проект?
2. Розкрийте зміст основних характеристик проекту.
3. Наведіть основні класифікації проектів.
4. Які особливості проектів ІС?
5. Наведіть загальну схему проекту як виду діяльності.
6. Що таке управління проектами?
7. Назвіть основні компоненти управління проектом.

8. Що таке трикутник обмежень в управлінні проектом?
9. Охарактеризуйте основні об'єкти управління проектом.
10. Наведіть склад процесів управління проектами. Як вони між собою пов'язані?
11. Охарактеризуйте процеси планування проекту.
12. Який зміст процесів виконання і контролю проекту?
13. Наведіть склад і зміст процесів аналізу проекту.
14. Що є процесами оперативного управління проектом?
15. Які сторони беруть участь у створенні проекту? Які їх функції?
16. Наведіть схеми організації проектування за участю чотирьох сторін. Які їх переваги і недоліки?
17. Назвіть організаційні форми управління в організаціях-розробниках. Розкрийте їх сутність.
18. Які принципи поділу праці в проектних колективах? У чому їх особливості?
19. Охарактеризуйте рольові кластери функцій проектної групи MSF.
20. Наведіть основні ролі проектної групи MSF.
21. Яка рольова структура проекту компанії IBM? Порівняйте ці ролі з проектною групою MSF.
22. Які існують внутрішні організаційні структури проектної групи, які їх особливості, склад виконавців та їх функції?
23. Які вимоги висуваються до колективів фахівців проекту реінжинірингу бізнес-процесів?
24. Наведіть організаційну структуру проекту реінжинірингу бізнес-процесів. У чому полягає сутність робіт його учасників?
25. Що розуміють під терміном "системний інтегратор"? Які функції виконують системні інтегратори?
26. Які типи рівнів рішень забезпечуються сучасним ринком програмних проектів?
27. Що означає термін "аутсорсинг" на ринку інформаційних послуг?
28. Яким чином розподіляються функції взаємодії користувача, проблемного програміста, системного програміста та системного інтегратора з програмно-технічним комплексом ІС?
29. Що таке проектна інтеграція? Чим вона відрізняється від системної інтеграції?
30. Які методи використовуються для формалізованого подання складу проектних робіт?
31. У чому суть використання методу діаграм Гантта?
32. Яка суть методу МПУ і методу критичного шляху?

33. Що таке мережний графік, який склад його компонентів і правила їх побудови?
34. Розкрити зміст основних термінів, що використовуються в методах МПУ.
35. Що таке ресурси в методах МПУ, які їх види? Що таке ресурсне календарне планування?
36. У яких напрямках може виконуватися аналіз можливості реалізувати проект?
37. За якими параметрами повинне здійснюватися управління проектом?
38. Які етапи виконуються в процесі застосування методу МПУ?
39. Які показники використовуються для розрахунку мережного графіка?
40. Що таке оптимізація мережного графіка?
41. Які існують методи узгодження мережного графіка із заданими термінами?
42. Що таке коефіцієнт напруженості?
43. Як здійснюється оптимізація за ресурсами?
44. Яка послідовність робіт з вибору інструментального засобу автоматизації управління проектуванням?
45. Наведіть склад чинників, що визначають вибір інструментального засобу управління проектом.
46. Які базові функціональні можливості повинен мати програмний засіб управління проектами?
47. Які на сьогодні існують найбільш популярні програмні засоби управління проектами? Наведіть їх порівняльний аналіз.

Рекомендована література

Основна

1. Анфилатов В. С. Системный анализ в управлении / В. С. Анфилатов, А. А. Емельянов, А. А. Кукушкин. – М.: Финансы и статистика, 2002. – 468 с.
2. Береза А. М. Основи створення інформаційних систем: Навч. посібник. – К.: КНЕУ, 2001. – 216 с.
3. Богданов В. В. Управление проектами в Microsoft Project. – СПб.: Питер, 2004. – 604 с.
4. Вигерс К. Разработка требований к программному обеспечению / Пер. англ. – М.: Изд.-торговый дом "Русская редакция", 2004. – 576 с.
5. Елиферов В. Г. Бизнес-процессы: Регламентация и управление:

Учебник / В. Г. Елиферов, В. В. Репин. – М.: ИНФРА-М, 2004. – 320 с.

6. Кальянов Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов. – М.: Горячая линия – Телеком, 2002. – 320 с.

7. Каменова М. Моделирование бизнеса. Методология ARIS / М. Каменова, А. Громов, М. Ферাপонтов, А. Шматолук. – М.: Весть-Мета Технология, 2001. – 328 с.

8. Лямец В. И. Системный анализ. Вводный курс / В. И. Лямец, А. Д. Тевяшев. – Харьков: ХНУРЭ, 2004. – 448 с.

9. Смирнова Г. Н. Проектирование экономических информационных систем: Учебник / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов; Под ред. Ю. Ф. Тельнова. – М.: Финансы и статистика, 2001. – 512 с.

10. Ушакова І. О. Основи системного аналізу об'єктів та процесів проектування: Навчальний засібник. Частина 1. – Харків: Вид. ХНЕУ, 2007. – 212 с.

11. Ушакова І. О. Системний аналіз та проектування систем обробки інформації. Конспект лекцій. – Харків: Вид. ХНЕУ, 2004. – 164 с.

12. Шафер Д. Ф. Управление программными проектами: достижение оптимального качества при минимуме затрат / Д. Ф. Шафер, Р. Т. Фатрел, Л. И. Шафер. – М.: Изд. дом "Вильямс", 2003. – 1136 с.

Додаткова

13. Годин В. В. Управление информационными ресурсами / В. В. Годин, И. К. Корнеев. – М.: ИНФРА - М, 2000. – 352 с.

14. ГОСТ 19.701-90. Схемы алгоритмов, данных, программ и систем. Условные обозначения и правила выполнения. – М.: Издательство стандартов, 1990. – 16 с.

15. ГОСТ 34.201-89. Виды, комплектность и обозначение документов при создании автоматизированных систем. – М.: Издательство стандартов, 1989. – 16 с.

16. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. – М.: Издательство стандартов, 1990. – 12 с.

17. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы. – М.: Издательство стандартов, 1990. – 24 с.

18. Гужва В. М. Інформаційні системи і технології на підприємствах: Навч. посібник. – К.: КНЕУ, 2001. – 400 с.

19. Гужва В. М. Інформаційні системи в міжнародному бізнесі: Навч. Посібник / В. М. Гужва, А. Г. Постевой. – К.: КНЕУ, 1999. – 164 с.

20. Дегтярев Ю. И. Системный анализ и исследование операций. – М.: Высшая школа, 1996. – 336 с.

21. Деордица Ю. С. Компьютерные технологии в маркетинге

- / Ю. С. Деордица, В. Т. Савченко. – Луганск: ВУГУ, 1998. – 240 с.
22. ДСТУ 2938-94. Системи оброблення інформації. Основні поняття. Терміни та визначення. – К.: Держстандарт України, 1995. – 32 с.
23. ДСТУ 2940-94. Системи оброблення інформації. Керування процесами оброблення даних. Терміни та визначення. – К.: Держстандарт України, 1995. – 28 с.
24. ДСТУ 2941-94. Системи оброблення інформації. Розробки систем. Терміни та визначення. – К.: Держстандарт України, 1995. – 20 с.
25. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Правила оформлення. – К.: Держкомстат України, 1995. – 28 с.
26. Єрьоміна Н. В. Банківські інформаційні системи: Навч. посібник. – К.: КНЕУ, 2000. – 272 с.
27. Інформаційні системи в економіці / За ред. В. С. Пономаренка. – К.: Академія, 2002. – 542 с.
28. Имери В. Как сделать бизнес в Internet. – К.: Диалектика, 1997. – 256 с.
29. Информационные системы в экономике. Учебник / Под ред. В. В. Дика. – М.: Финансы и статистика, 1996. – 272 с.
30. Информационные технологии в бизнесе: Энциклопедия.: Пер. с англ. / Под ред. М. Желены. – СПб.: Питер, 2002. – 1 120 с.
31. Карминский А. М. Информатизация бизнеса / А. М. Карминский, А. С. Карминский, В. П. Нестеров, Б. В. Черников. – М.: Финансы и статистика, 2004. – 624 с.
32. Лагоша Б. А. Основы системного анализа / Б. А. Лагоша, А. А. Емельянов – М.: Издательство МЭСИ, 1998. – 108 с.
33. Месарович М. Математическая теория систем. – М.: Мир, 1971. – 316 с.
34. Мороз А. И. Курс теории систем. – М.: Высшая школа, 1987. – 320 с.
35. Нечипоренко В. И. Структурный анализ систем. – М.: Сов. радио, 1977. – 96 с.
36. Одрин М. В. Морфологический анализ систем. / М. В. Одрин, С. С. Карташов – К.: Наукова думка, 1977. – 124 с.
37. Ойхман Е. Г. Реинжиниринг бизнеса: реинжиниринг организаций и информационные технологии / Е. Г. Ойхман, Е. В. Попов. – М.: Финансы и статистика, 1997. – 336 с.
38. Орлов П. І. Інформаційні системи та технології в управлінні, освіті, бібліотечній справі / П. І. Орлов, О. М. Луганський. – Харків: Вид. "Прометей-Прес", 2002. – 292 с.
39. Острейковский В. А. Теория систем: Учебник для вузов. – М.: Высшая школа, 1997. – 240 с.

40. Перегудов Ф. И. Введение в системный анализ / Ф. И. Перегудов, Ф. П. Тарасенко. – М.: Высшая школа, 1989. – 368 с.
41. Пономаренко В. С. Проектування автоматизованих економічних інформаційних систем. Навч. Посібник / В. С. Пономаренко, О. І. Пушкар, Ю. І. Коваленко. – К.: ІЗМН, 1996. – 312 с.
42. Первичная документация / Сост. О. Пироженко. – 3-е изд., изм. и доп. – Харьков: Фактор, 2003. – 220 с.
43. Петров В. Н. Информационные системы. – СПб.: Питер, 2002. – 688 с.
44. Писаревська Т. А. Інформаційні системи в управлінні трудовими ресурсами. – К.: КНЕУ, 1997. – 252 с.
45. Пінчук Н. С. Інформаційні системи і технології в маркетингу: Навч. посібник / Н. С. Пінчук, Г. П. Галузинський, Н. С. Орленко. – К.: КНЕУ, 1999. – 328 с.
46. РД 50-34.698-90. Руководящий документ по стандартизации. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов. – М.: Издательство стандартов, 1991. – 40 с.
47. Саттон М. Корпоративный документооборот: принципы, технологии, методология внедрения. – СПб.: Азбука, 2002. – 432 с.
48. Ситник В. Ф. Основи інформаційних систем / В. Ф. Ситник, Т. А. Писаревська, Н. В. Єрьоміна, О. С. Краєва. – К.: КНЕУ, 1997. – 252 с.
49. Ситник В. Ф. Телекомунікації в бізнесі: Навч.-метод. посібник для самост. вивч. дисц. / В. Ф. Ситник, І. А. Козак – К.: КНЕУ, 1999. – 204 с.
50. Уткин Э. А. Бизнес-реинжиниринг. – М.: Ассоциация авторов и издателей "Тандем"; Изд. "ЭКМОС", 1998. – 224 с.
51. Ушакова І. О. Інформаційні системи і технології в статистиці. – Харків: Вид. ХНЕУ, 2006. – 164 с.
52. Ушакова І. О. Системи обробки статистичної інформації. – Харків: Вид. ХНЕУ, 2005. – 160 с.
53. Хотяшов Э. Н. Проектирование машинной обработки экономической информации. – М.: Финансы и статистика, 1987. – 248 с.
54. Черняк Ю. И. Системный анализ в управлении экономикой. – М.: Экономика, 1975. – 180 с.
55. Шрейдер Ю. А. Системы и модели / Ю. А. Шрейдер, А. А. Шаров. – М.: Радио и связь, 1982. – 84 с.
56. Экономическая информатика / Под ред. проф. В. В. Евдокимова. – СПб.: Питер, 1997. – 592 с.

Ресурси мережі Internet

57. Інтернет-журнал Link Львівського сайту інформаційних технологій // ITEL. <http://itel.netfirms.com/>
58. Історія розвитку інформаційних технологій в Україні // http://www.icfcst.kiev.ua/MUSEUM/IT_u.html/
59. Портал дистанційного консультування // http://www.dist-cons.ru/con_mod.asp/
60. Сайт компанії Gartner, Inc. (NYSE: IT) // <http://www.gartner.com/>
61. Сайт компанії Interface: Internet and software company // <http://www.interface.ru/>
62. Сайт компанії Microsoft // <http://www.microsoft.com/>
63. Справочник економічних термінів. Сайт "Управління фірмою" // <http://upr.org.ru/DictR.htm/>
64. Центр інформаційних технологій // <http://www.citmgu.ru/>

**Визначення термінів стосовно розробки та застосування ПЗ, подані
в стандарті ДСТУ 3918-1999**

Назва терміна	Визначення терміна
1	2
Аудит (audit)	Ревізія, що здійснюється уповноваженою особою з метою забезпечення незалежної оцінки програмних продуктів та процесів, щоб оцінити їх відповідність вимогам
Базис (baseline)	Офіційно схвалена версія елемента конфігурації, незалежна від середовища, формально розроблена та виправлена впродовж значного часу в рамках життєвого циклу елемента конфігурації
Безпека (security)	Захист інформації та даних у такий спосіб, щоб неуповноважена особа не могла їх прочитати або змінити, але щоб не перешкоджала доступові до них з боку уповноважених осіб чи організацій
Валідація (validation)	<p>Підтвердження того, що окремі вимоги щодо визначеного передбаченого використання виконуються і воно здійснюється шляхом перевірки та забезпечення об'єктивних доказів.</p> <p>Примітка 1. У процесі проектування та розроблення валідація полягає у процесі перевірки продукту для встановлення відповідності потребам користувача</p> <p>Примітка 2. Валідація виконується, як правило, над кінцевим продуктом у заданих умовах функціонування. Потреба в ній може виникнути на більш ранніх етапах.</p> <p>Примітка 3. Термін "валідований" використовується для надання відповідного статусу.</p> <p>Примітка 4. Багаторазова валідація може здійснюватись у разі наявності різних передбачених використань (ISO8402, 2.18)</p>
Верифікація (verification)	<p>Підтвердження виконання заданих вимог, що здійснюється шляхом перевірки та забезпечення об'єктивних доказів.</p> <p>Примітка 1. У процесі проектування та розроблення верифікація полягає в процесі перевірки результатів певної діяльності для визначення відповідності вимогам, встановленим щодо цієї діяльності.</p> <p>Примітка 2. Термін "верифікація" використовується для надання відповідного статусу (ISO 8402:1994, 2.17)</p>

1	2
Версія (version)	Примірник елемента, що підлягає ідентифікації. Примітка. Модифікація версії програмного продукту, результатом якої є нова версія потребує дії щодо керування конфігурацією
Вилучення (retirement)	Припинення активної підтримки з боку організації, що здійснює експлуатацію та супровід, часткова чи повна заміна на нову систему або введення в дію оновленої системи
Готовий для використання продукт (off-shell product)	Вже розроблений та доступний продукт, що може використовуватись у початковому вигляді ("як є") або із модифікацією
Елемент конфігурації (configuration item)	Об'єкт (сутність) у рамках конфігурації, який задовольняє функцію кінцевого користування і може бути однозначно ідентифікований з певної точки зору (у вказаному значенні, щодо певної базової точки)
Елемент, що не передається (non-deliverable item)	Апаратні чи програмні продукти, які не потребують передачі згідно з контрактом, але які можуть застосовуватись під час розроблення програмного продукту
Забезпечення якості (quality assurance)	Всі дії, що плануються та регулярно здійснюються в рамках системи якості і належним чином демонструються з метою забезпечення достатньої впевненості в тому, що об'єкт буде задовольняти вимоги щодо якості. Примітка 1. Забезпечення якості може мати як внутрішню, так і зовнішню мету: а) внутрішнє забезпечення якості. У межах організації, забезпечення якості передбачає надання впевненості керівництву; б) зовнішнє забезпечення якості. За наявності відповідного контракту забезпечення якості передбачає надання впевненості клієнтові або іншим організаціям чи особам. Примітка 2. Між деякими діями з керування якістю та забезпечення якості існує взаємозв'язок. Примітка 3. Якщо вимоги щодо якості не відображають повністю потреби користувача, забезпечення якості може не передбачати достатньої впевненості (ISO 8402, 3.5)
Замовник (customer)	Організація, що замовляє або отримує систему, програмний продукт чи послугу від постачальника Примітка. Термін "замовник" є синонімом покупця, клієнта, власника або користувача
Замовлення (acquisition)	Процес одержання системи, програмного продукту або програмної послуги
Запит щодо пропозиції [тендер] (request for proposal [tender])	Документ, що використовується замовником як засіб для оголошення своїх намірів потенційним учасникам тендеру щодо замовлення певної системи, програмного продукту чи програмної послуги
Кваліфікація (qualification)	Процес демонстрації здатності об'єкта виконувати задані вимоги (ISO 8402, 2.13)

Продовження табл. А.1

с	2
Кваліфікаційні вимоги (qualification requirements)	Множина критеріїв чи вимог, які повинні задовольнятися для того, щоб програмний продукт можна було кваліфікувати як такий, що відповідає специфікаціям на нього і готовий для використання в його цільовому середовищі
Кваліфікаційне випробування (qualification testing)	Випробування, яке проводить розробник і засвідчує замовник (відповідним чином) для демонстрації того, що програмний продукт відповідає специфікаціям на нього і готовий для використання в його цільовому середовищі
Контракт (contract)	Двостороння угода, якій, як правило, надано юридичної сили, або подібна внутрішня угода, виключно у межах організації, щодо надання програмних послуг або постачання, розроблення, виробництва, експлуатації чи супроводу програмного продукту
Користувач (user)	Особа або організація, що використовує діючу систему для виконання заданої функції. Примітка. Користувач може виконувати інші ролі, наприклад, замовника, розробника, супроводжувача
Модель життєвого циклу (life cycle model)	Концептуальна структура, що включає процеси, дії та завдання, які стосуються розроблення, експлуатації та супроводу програмного продукту, і охоплює життєвий цикл системи, починаючи з визначення вимог до неї і закінчуючи припиненням її використання
Моніторинг (monitoring)	Перевірка стану діяльності постачальника та її результат з боку замовника або третьої сторони
Оператор (operator)	Організація, що експлуатує систему (оперує з системою, керує системою)
Оцінювання (evaluation)	Систематичне визначення ступеня відповідності об'єкта (сутності) заданим щодо нього критеріям
Постачальник (supplier)	Організація, що укладає контракт з замовником на постачання системи, програмного продукту чи програмної послуги згідно з умовами контракту. Примітка 1. Термін "постачальник" є синонімом підрядника, виробника, продавця або торговця Примітка 2. Замовник може визначити частину своєї організації як підрядника
Програмна послуга (software service)	Виконання діяльності, роботи або обов'язків, пов'язаних із програмним продуктом, наприклад таких, як його розроблення, супровід та експлуатація

1	2
Програмний модуль (software unit)	Фрагмент коду, що може бути скопільований окремо
Програмний продукт (software product)	Набір комп'ютерних програм, процедур та, можливо, пов'язаних із ними документації та даних
Програмно-апаратні засоби (firmware)	Поєднання апаратного пристрою та машинних інструкцій або комп'ютерних даних, що розміщуються як програмний засіб типу "тільки для читання" в апаратному пристрої. Програмний засіб не можна швидко змінювати програмним шляхом
Процес (process)	Множина взаємопов'язаних дій, які перетворюють входи на виходи. Примітка. Термін "дії" охоплює використання ресурсів (ISO 8402, 1.2)
Редакція (release)	Окрема версія елемента конфігурації, що надається з певною метою (наприклад, тестова редакція)
Розробник (developer)	Організація, що провадить дії з розроблення (включно з аналізом вимог, проектуванням, випробуваннями під час приймання) в межах життєвого циклу програмного забезпечення
Система (system)	Інтегрована структура, що складається з одного чи більшої кількості процесів, компонентів апаратного забезпечення, засобів та персоналу, що забезпечує можливість задоволення встановленої потреби або цільової функції
Супроводжувач (maintainer)	Організація, що здійснює діяльність із супроводу
Тестове покриття (test coverage)	Ступінь, до якого набір тестових даних випробовує вимоги до системи або програмного продукту. (Міра, що характеризує здатність тестових даних випробовувати вимоги до системи або програмного продукту)
Тестопридатність (testability)	Ступінь можливості розроблення об'єктивного та здійсненого тесту для встановлення відповідності вимогам
Угода (agreement)	Визначення термів та умов, за яких будуть здійснюватися робочі стосунки
Формулювання роботи (statement of work)	Документ, що використовується замовником як засіб опису та специфікації завдань, які слід виконати згідно з контрактом

Зміст

Вступ	3
Модуль 3. Теоретичні основи проектування ІС	4
8. Методологія проектування ІС	4
8.1. Технологія проектування ІС	4
8.2. Формалізація технології проектування ІС	10
8.3. Технологічна мережа проектування	12
8.4. Методи і засоби проектування	14
9. Програмна інженерія	20
9.1. Поняття життєвого циклу програмного продукту	20
9.1.1. Життєвий цикл промислового виробу	20
9.1.2. Визначення програмної інженерії	21
9.1.3. Особливості життєвого циклу програмного продукту	24
9.2. Стандарти життєвого циклу програмного забезпечення	26
9.2.1. Стандарти і сертифікація	26
9.2.2. Види стандартів	28
9.2.3. Стандарти процесів життєвого циклу програмного забезпечення	31
9.2.4. Стандарти процесів ЖЦ ПЗ, розроблені ISO	32
9.2.5. Стандарти зрілості можливостей організації, розроблені SEI	44
9.3. Традиційні моделі життєвого циклу програмного забезпечення	49
9.3.1. Поняття моделі життєвого циклу програмного забезпечення	49
9.3.2. Каскадна модель	50
9.3.3. V-подібна модель	55
9.3.4. Модель швидкого прототипування	57
9.3.5. Модель швидкої розробки ПЗ (RAD)	62
9.3.6. Інкрементна (покрокова) модель	65
9.3.7. Спіральна модель	68
9.4. Моделі життєвого циклу програмного продукту на основі індустріальних технологій	73
9.4.1. Важкі і гнучкі технології проектування	73
9.4.2. Методологія RUP	75

9.4.3. Методологія Microsoft Solution Framework	81
9.4.4. eXtreme Programming та інші гнучкі методології	87
9.5. Аналіз вимог до програмного забезпечення	95
10. Канонічне проектування ІС	10
10.1. Технологія передпроектного обстеження	4
10.1.1. Склад стадій і етапів канонічного проектування ІС	10
10.1.2. Об'єкти обстеження в процесі створення ІС	4
10.1.3. Склад і зміст робіт на етапі "Збір матеріалів обстеження"	10
10.1.4. Методи проведення обстеження	4
10.1.5. Програма обстеження	10
10.1.6. Проведення збору і формалізації матеріалів обстеження	8
10.1.7. Склад і зміст робіт на етапі "Аналіз матеріалів обстеження"	10
10.1.8. Техніко-економічне обґрунтування	9
10.1.9. Технічне завдання	11
10.2. Технологія техноробочого проектування	0
10.2.1. Склад і зміст робіт на етапі технічного проектування	11
10.2.2. Постановка комплексу завдань	4
10.2.3. Склад і зміст робіт на етапі робочого проектування	11
10.2.4. Зміст проектної документації на інформаційні системи	6
10.3. Технологія впровадження, експлуатації і супроводу проекту	12
10.3.1. Склад і зміст робіт на стадії впровадження проекту	3
10.3.2. Склад і зміст робіт на стадії "Експлуатація і супровід проекту"	12
10.3.3. Приймально-здавальна документація на ІС	4
	7
	12
	8

	13
	1
	13
	2
	13
	4
	134
	13
	6
	13
	8
Модуль 4. Індустріальні методи проектування ІС	14
11. Підходи до проектування ІС	0
11.1. Методології моделювання предметної області	14
11.1.1. Система моделей предметної області	0
11.1.2. Рівні деталізації проблемної області	14
11.2. Структурний підхід до проектування ІС	0
11.2.1. Сутність структурного аналізу	14
11.2.2. Методології структурного підходу	0
	14
	4
	14
	8
	14
	8
	14
	9

11.2.3. Методологія функціонального моделювання SADT	15
11.2.4. Методології моделювання потоків даних	2
11.2.5. Методології моделювання даних	15
11.2.6. Методологія моделювання потоків робіт IDEF3	7
11.2.7. Структурне проектування	16
11.3. Об'єктний підхід до проектування ІС	0
11.3.1. Сутність об'єктного підходу	16
	1
	16
	2
	16
	3
	16
	3
	16
11.3.2. Стандарти об'єктного проектування	16
11.3.3. Технологія об'єктно-орієнтованого проектування	6
12. Комбінований процесно-орієнтований підхід до проектування ІС	17
	2
12.1. Передумови виникнення підходу	
12.2. Основні поняття процесного підходу	17
12.3. Основи реінжинірингу бізнес-процесів	7
12.3.1. Реінжиніринг бізнесу як основа сучасного менеджменту організації	17
	7
12.3.2. Фази реінжинірингу бізнес-процесів	17
12.4. Моделювання бізнес-процесів	8
12.4.1. Інструментальні засоби моделювання бізнес-процесів	18
	2
12.4.2. Опис бізнес-процесів у нотації ARIS	
	18
	2
	18
	6
	19
	4
	19

4
20
0

13. Типове проектування ІС	20
13.1. Основні поняття і види методів типового проектування ІС	8
13.1.1. Види методів типового проектування	20
13.1.2. Елементний метод типового проектування	8
13.1.3. Підсистемний метод типового проектування	20
13.1.4. Об'єктний метод типового проектування	8
13.2. Параметрично-орієнтоване проектування ІС	20
13.2.1. Основні потоки і компоненти ППП	6
13.2.2. Технологія проектування на основі параметричного настроювання ППП	21
	2
	21
	2
	21
	3
	21
	3
	21
	5

13.3. Модельно-орієнтоване проектування ІС	22
13.3.1. Конфігурація ІС на основі модельно-орієнтованої технології проектування	0
	22
	0
13.3.2. Компоненти моделі організації	22
13.3.3. Модель функцій	2
13.3.4. Модель процесів	22
13.3.5. Модель правил	4
13.3.6. Моделі об'єктів (даних)	22
	8
	22
	9
	23
	1

13.3.7. Модель організаційної структури	23
13.3.8. Технологія модельно-орієнтованого проектування	1
14. Управління проектуванням ІС	23
14.1. Особливості проектної діяльності	5
14.2. Основні складові управління проектами	23
14.3. Процеси управління проектами	8
14.4. Зовнішні структури організації робіт із проектування ІС	23
14.5. Організаційні форми управління проектами	8
14.6. Поділ праці у проектних колективах	24
14.7. Функціональні ролі в колективі розробників	1
14.8. Внутрішні організаційні структури проектної групи	24
14.9. Організаційні форми реінжинірингу бізнес-процесів	6
14.10. Сучасний ринок програмних проектів	25
14.10.1. Системна інтеграція	2
14.10.2. Проектна інтеграція	25
14.11. Планування і контроль проектних робіт	7
14.11.1. Методи планування та управління проектами і ресурсами	25
14.11.2. Технологія застосування методу МПУ для розробки проекту ІС	8
14.11.3. Розрахунок основних показників мережного графіка	26
14.11.4. Оптимізація мережного графіка	0
14.11.5. Інструментальні засоби управління проектами	26
	4
	26
	8
	27
	1
	27
	1
	27
	6
	27
	7
	27

7

28

1

28

3

28

6

28

9

Рекомендована література

30

0

Додаток

30

4

НАВЧАЛЬНЕ ВИДАННЯ

Ушакова Ірина Олексіївна

ОСНОВИ СИСТЕМНОГО АНАЛІЗУ ОБ'ЄКТІВ І ПРОЦЕСІВ КОМП'ЮТЕРИЗАЦІЇ

Навчальний посібник

Частина 2

Відповідальний за випуск **Пономаренко В. С.**

Відповідальний редактор **Сєдова Л. М.**

Редактор **Новицька О. С.**

Коректор **Новицька Л. М.**

План 2008 р. Поз. №47-П.

Підп. до друку

Формат 60 × 90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. __,__. Обл.-вид. арк. __,__. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ, 61001, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк №481 від 13.06.2001 р.*

Ушакова І. О.

**ОСНОВИ СИСТЕМНОГО АНАЛІЗУ ОБ'ЄКТІВ І ПРОЦЕСІВ
КОМП'ЮТЕРИЗАЦІЇ**

Навчальний посібник

Частина 2