

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра інформаційно-обчислювальних систем і управління

## **КОНСПЕКТ ЛЕКЦІЙ**

з дисципліни «**Моделі, технології проектування та управління ІС**»

ТЕРНОПІЛЬ 2015

## ЗМІСТ

1. Основи методології проектування ІС .....	4
1.1. Життєвий цикл ПЗ ІС .....	4
1.2. Моделі життєвого циклу ПЗ .....	5
1.3. Методології і технології проектування ІС .....	7
1.3.1. Загальні вимоги до методології і технології .....	7
1.3.2. Методологія RAD .....	10
2. Структурний підхід до проектування ІС .....	14
2.1. Суть структурного підходу .....	14
2.2. Методологія функціонального моделювання SADT .....	15
2.2.1. Склад функціональної моделі .....	15
2.2.2. Ієрархія діаграм .....	16
2.2.3. Типи зв'язків між функціями .....	19
2.3. Моделювання потоків даних (процесів) .....	23
2.3.1. Зовнішня суть .....	23
2.3.2. Системи і підсистеми .....	23
2.3.3. Процеси .....	24
2.3.4. Накопичувачі даних .....	24
2.3.5. Потоки даних .....	25
2.3.6. Побудова ієрархії діаграм потоків даних .....	25
2.4. Моделювання даних .....	27
2.4.1. Case-метод Баркера .....	27
2.4.2. Методологія IDEF1 .....	32
2.4.3. Підхід, що використовується в CASE-засобі Vantage Team Builder .....	34
2.5. Приклад використання структурного підходу .....	36
2.5.1. Опис предметної області .....	36
2.5.2. Організація проекту .....	37
3. Програмні засоби підтримки життєвого циклу ПЗ .....	45
3.1. Методології проектування ПЗ як програмні продукти .....	45
3.1.1. Методологія DATARUN .....	45
3.1.2. Інструментальний засіб SE Companion .....	50
3.2. CASE-засоби. Загальна характеристика і класифікація .....	52
4. Технологія впровадження CASE-засобів .....	55
4.1. Визначення потреб в CASE-засобах .....	55
4.1.1. Аналіз можливостей організації .....	56
4.1.2. Визначення організаційних потреб .....	58
4.1.3. Аналіз ринку CASE-засобів .....	60
4.1.4. Визначення критеріїв успішного впровадження .....	61
4.1.5. Розробка стратегії впровадження CASE-засобів .....	61
4.2. Оцінка і вибір CASE-засобів .....	64
4.2.1. Загальні відомості .....	64
4.2.2. Процес оцінки .....	65
4.2.3. Процес вибору .....	67

4.2.4. Критерії оцінки і вибору.....	69
4.2.5. Приклад підходу до визначення критеріїв вибору CASE-засобів ..	77
4.3. Виконання пілотного проекту .....	81
4.4. Перехід до практичного використання CASE-засобів.....	89
5. Характеристики CASE-засобів .....	96
5.1. Silverrun+JAM .....	96
5.1.1. Silverrun .....	96
5.1.2. JAM .....	98
5.2. Vantage Team Builder (Westmount I-CASE) + Uniface.....	103
5.2.1. Vantage Team Builder (Westmount I-CASE) .....	103
5.2.2. Uniface .....	106
5.3. Designer/2000 + Developer/2000.....	107
5.4. Локальні засоби (ERwin, VPwin, S-Designor, CASE.Аналітик) .....	109
5.5. Об'єктно-орієнтовані CASE-засоби (Rational Rose).....	111
5.6. Допоміжні засоби підтримки життєвого циклу ПЗ.....	113
5.6.1. Засоби конфігураційного управління.....	113
5.6.2. Засоби документування .....	117
5.6.3. Засоби тестування .....	118
5.7. Приклади комплексів CASE-засобів .....	119
6. Моделі управління підприємствами в інформаційних системах.....	121
6.1. Загальна характеристика систем автоматизації управління ресурсами підприємства.....	121
6.2. MRP-система (планування матеріальних ресурсів) .....	124
6.3. MRP II-система (планування виробничих ресурсів).....	128
6.4. ERP-система (планування ресурсів підприємства) .....	130
6.4.1. Застосування системи Microsoft XAL на світових ринках .....	139
6.4.2. Основні модулі ERP-системи компанії BAAN IV .....	156
6.4.3. Концепція ERP II.....	162
6.5. CSRP-система (планування ресурсів, синхронізоване зі споживачем) .....	173
Література .....	179

# **1. Основи методології проектування ІС**

## **1.1. Життєвий цикл ПЗ ІС**

Одним з базових понять методології проектування ІС є поняття життєвого циклу її програмного забезпечення (ЖЦ ПЗ). ЖЦ ПЗ – це безперервний процес, який починається з моменту ухвалення рішення про необхідність його створення і закінчується у момент його повного вилучення з експлуатації.

Основним нормативним документом, регламентуючим ЖЦ ПЗ, є міжнародний стандарт ISO/IEC 12207 [5] (ISO – International Organization of Standardization – Міжнародна організація по стандартизації, IEC – International Electrotechnical Commission - Міжнародна комісія з електротехніки). Він визначає структуру ЖЦ, що містить процеси, дії і завдання, які повинні бути виконані під час створення ПЗ.

Структура ЖЦ ПЗ за стандартом ISO/IEC 12207 базується на трьох групах процесів:

- основні процеси ЖЦ ПЗ (придбання, постачання, розробка, експлуатація, супровід);
- допоміжні процеси, що забезпечують виконання основних процесів (документування, управління конфігурацією, забезпечення якості, верифікація, атестація, оцінка, аудит, вирішення проблем);
- організаційні процеси (управління проектами, створення інфраструктури проекту, визначення, оцінка і поліпшення самого ЖЦ, навчання).

Розробка включає всі роботи із створення ПЗ і його компонент відповідно до заданих вимог, включаючи оформлення проектної і експлуатаційної документації, підготовку матеріалів, необхідних для перевірки працездатності і відповідної якості програмних продуктів, матеріалів, необхідних для організації навчання персоналу і так далі. Розробка ПЗ включає, як правило, аналіз, проектування і реалізацію (програмування).

Експлуатація включає роботи по впровадженню компонентів ПЗ в експлуатацію, зокрема конфігурацію бази даних і робочих місць користувачів, забезпечення експлуатаційною документацією, проведення навчання персоналу і так далі, і безпосередньо експлуатацію, зокрема локалізацію проблем і усунення причин їх виникнення, модифікацію ПЗ в рамках встановленого регламенту, підготовку пропозицій по вдосконаленню, розвитку і модернізації системи.

Управління проектом пов'язане з питаннями планування і організації робіт, створення колективів розробників і контролю за термінами і якістю виконуваних робіт. Технічне і організаційне забезпечення проекту включає вибір методів і інструментальних засобів для реалізації проекту, визначення методів опису проміжних станів розробки, розробку методів і засобів випробувань ПЗ, навчання персоналу і тому подібне. Забезпечення якості проекту пов'язане з проблемами верифікації, перевірки і тестування ПЗ.

Верифікація - це процес визначення того, чи відповідає поточний стан розробки, досягнутий на даному етапі, вимогам цього етапу. Перевірка дозволяє оцінити відповідність параметрів розробки з початковими вимогами. Перевірка частково співпадає з тестуванням, яке пов'язане з ідентифікацією відмінностей між дійсними і очікуваними результатами і оцінкою відповідності характеристик ПЗ початкових вимогах. В процесі реалізації проекту важливе місце займають питання ідентифікації, опису і контролю конфігурації окремих компонентів і всієї системи в цілому.

Управління конфігурацією є одним з допоміжних процесів, що підтримують основні процеси життєвого циклу ПЗ, перш за все процеси розробки і супроводу ПЗ. При створенні проектів складних ІС, що складаються з багатьох компонентів, кожен з яких може мати різновиди або версії, виникає проблема обліку їх зв'язків і функцій, створення уніфікованої структури і забезпечення розвитку всієї системи. Управління конфігурацією дозволяє організувати, систематично враховувати і контролювати внесення змін ПЗ на всіх стадіях ЖЦ. Загальні принципи і рекомендації конфігураційного обліку, планування і управління конфігураціями ПЗ відбиті в проекті стандарту ISO 12207-2 [5].

Кожен процес характеризується певними завданнями і методами їх рішення, початковими даними, отриманими на попередньому етапі, і результатами. Результатами аналізу, зокрема, є функціональні моделі, інформаційні моделі і відповідні ним діаграми. ЖЦ ПЗ носить ітераційний характер: результати чергового етапу часто викликають зміни в проектних рішеннях, вироблених на раніших етапах.

## **1.2. Моделі життєвого циклу ПЗ**

Стандарт ISO/IEC 12207 не пропонує конкретну модель ЖЦ і методи розробки ПЗ (під моделлю ЖЦ розуміється структура, що визначає послідовність виконання і взаємозв'язку процесів, дій і завдань, що виконуються впродовж ЖЦ. Модель ЖЦ залежить від специфіки ІС і специфіки умов, в яких остання створюється і функціонує). Його регламенти є загальними для будь-яких моделей ЖЦ, методологій і технологій розробки. Стандарт ISO/IEC 12207 описує структуру процесів ЖЦ ПЗ, але не конкретизує в деталях, як реалізувати або виконати дії і завдання, включені в ці процеси.

До теперішнього часу найбільшого поширення набули наступні дві основні моделі ЖЦ:

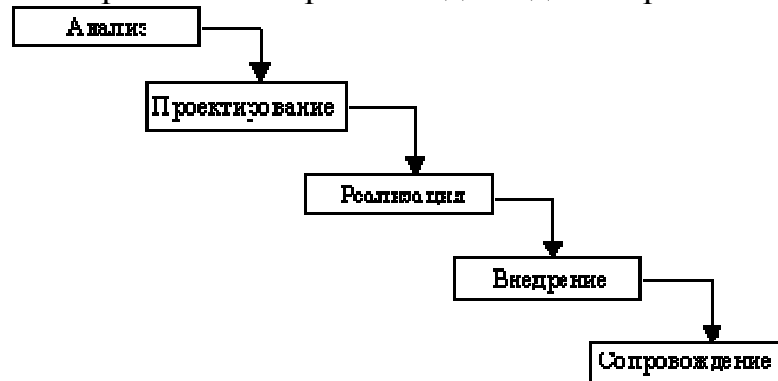
- каскадна модель (70-85 pp.);
- спіральна модель (86-90 pp.).

У спочатку існуючих однорідних ІС кожне застосування було єдиним цілим. Для розробки такого типу застосувань застосовувався каскадний спосіб. Його основною характеристикою є розбиття всієї розробки на етапи, причому перехід з одного етапу на наступний відбувається тільки після того, як буде повністю завершена робота на поточному (рис. 1.1). Кожен етап завершується випуском повного комплексу документації, достатньої для

того, щоб розробка могла бути продовжена іншою командою розробників.

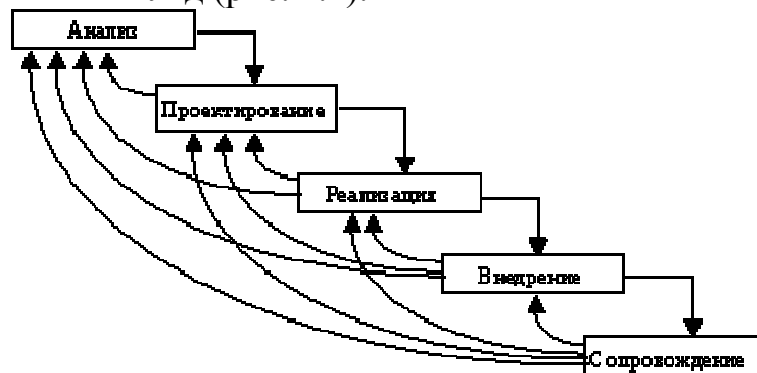
Позитивні сторони застосування каскадного підходу полягають в наступному [2]:

- на кожному етапі формується закінчений набір проектної документації, що відповідає критеріям повноти і узгодженості;
- виконувані в логічній послідовності етапи робіт дозволяють планувати терміни завершення всіх робіт і відповідні витрати.



Мал. 1.1. Каскадна схема розробки ПЗ

Каскадний підхід добре зарекомендував себе при побудові ІС, для яких на самому початку розробки можна достатньо точно і повно сформулювати всі вимоги, з тим щоб надати розробникам свободу реалізувати їх якнайкраще з технічної точки зору. У цю категорію потрапляють складні розрахункові системи, системи реального часу і інші подібні завдання. Проте, в процесі використання цього підходу виявилися ряд його недоліків, викликаних перш за все тим, що реальний процес створення ПЗ ніколи повністю не укладався в таку жорстку схему. В процесі створення ПЗ постійно виникала потреба в поверненні до попередніх етапів і уточненні або перегляді раніше ухвалених рішень. В результаті реальний процес створення ПЗ приймав наступний вигляд (рис. 1.2):



Мал. 1.2. Реальний процес розробки ПЗ по каскадній схемі

Основним недоліком каскадного підходу є істотне запізнювання з отриманням результатів. Узгодження результатів з користувачами проводиться тільки в крапках, що плануються після завершення кожного етапу робіт, вимоги до ІС "заморожені" у вигляді технічного завдання на весь час її створення. Таким чином, користувачі можуть внести свої зауваження

тільки після того, як робота над системою буде повністю завершена. У разі неточного викладу вимог або їх зміни протягом тривалого періоду створення ПЗ, користувачі отримують систему, що не задовольняє їх потребам. Моделі (як функціональні, так і інформаційні) об'єкту, що автоматизується, можуть застаріти одночасно з їх твердженням.

Для подолання перерахованих проблем була запропонована спіральна модель ЖЦ [10] (рис. 1.3), що робить упор на початкові етапи ЖЦ: аналіз і проектування. На цих етапах та, що реалізовується технічних рішень перевіряється шляхом створення прототипів. Кожен виток спіралі відповідає створенню фрагмента або версії ПЗ, на ній уточнюються цілі і характеристики проекту, визначається його якість і плануються роботи наступного витка спіралі. Таким чином заглиблюються і послідовно конкретизуються деталі проекту і в результаті вибирається обґрунтований варіант, який доводиться до реалізації.

Розробка ітераціями відображає об'єктивно існуючий спіральний цикл створення системи. Неповне завершення робіт на кожному етапі дозволяє переходити на наступний етап, не чекаючи повного завершення роботи на поточному. При ітеративному способі розробки бракуючу роботу можна буде виконати на наступній ітерації. Головне ж завдання - щонайшвидше показати користувачам системи працездатний продукт, тим самим активізуючи процес уточнення і доповнення вимог.

Основна проблема спірального циклу - визначення моменту переходу на наступний етап. Для її вирішення необхідно ввести тимчасові обмеження на кожен з етапів життєвого циклу. Перехід здійснюється відповідно до плану, навіть якщо не вся запланована робота закінчена. План складається на основі статистичних даних, отриманих в попередніх проектах, і особистого досвіду розробників.

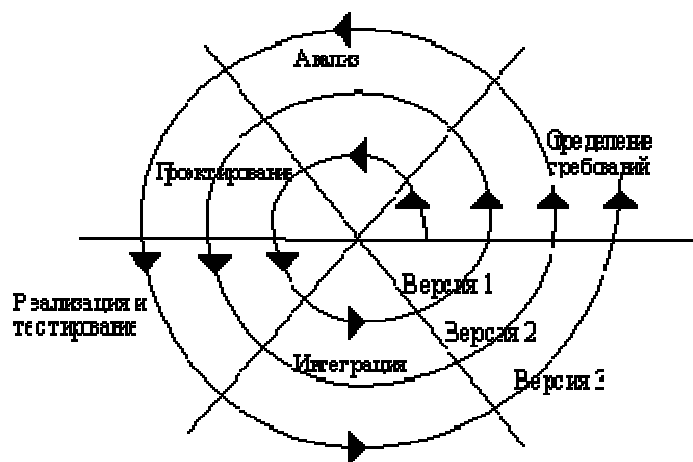


Рис 1.3. Спіральна модель ЖЦ

### 1.3. Методології і технології проектування ІС

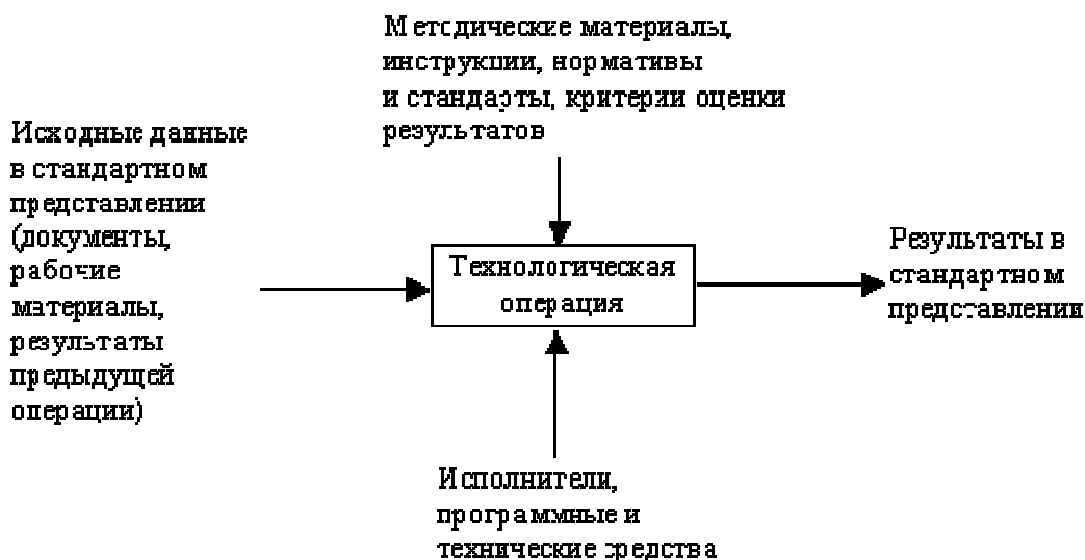
#### 1.3.1. Загальні вимоги до методології і технології

Методології, технології і інструментальні засоби проектування (CASE-засоби) складають основу проекту будь-якої ІС. Методологія реалізується

через конкретні технології і стандарти, що підтримують їх, методики і інструментальні засоби, які забезпечують виконання процесів ЖЦ.

Технологія проектування визначається як сукупність три складових:

- покрокової процедури, що визначає послідовність технологічних операцій проектування (мал. 1.4);
- критеріїв і правил, використовуваних для оцінки результатів виконання технологічних операцій;
- нотацій (графічних і текстових засобів), використовуваних для опису проектованої системи.



Мал. 1.4. Представлення технологічної операції проектування

Технологічні інструкції, складові основний зміст технології, повинні складатися з опису послідовності технологічних операцій, умов, залежно від яких виконується та або інша операція, і описів самих операцій.

Технологія проектування, розробки і супроводу ІС повинна задовольняти наступним загальним вимогам:

- технологія повинна підтримувати повний ЖЦ ПЗ;
- технологія повинна забезпечувати гарантоване досягнення цілей розробки ІС із заданою якістю і у встановлений час;
- технологія повинна забезпечувати можливість виконання великих проектів у вигляді підсистем (тобто можливість декомпозиції проекту на складові частини, що розробляються групами виконавців обмеженої чисельності з подальшою інтеграцією складових частин). Досвід розробки великих ІС показує, що для підвищення ефективності робіт необхідно розбити проект на окремих слабо зв'язані по даним і функціям підсистеми. Реалізація підсистем повинна виконуватися окремими групами фахівців. При цьому необхідно забезпечити координацію ведення загального проекту і виключити дублювання результатів робіт кожної проектною групою, яке може виникнути через наявність загальних даних і функцій;
- технологія повинна забезпечувати можливість ведення робіт по проектуванню окремих підсистем невеликими групами (3-7 чоловік). Це обумовлено принципами керування колективом і підвищення продуктивності



за рахунок мінімізації числа зовнішніх зв'язків;

– технологія повинна забезпечувати мінімальний час отримання працездатної ІС. Мова йде не про терміни готовності всієї ІС, а про терміни реалізації окремих підсистем. Реалізація ІС в цілому в короткі терміни може зажадати залучення великого числа розробників, при цьому ефект може опинитися нижче, ніж при реалізації в коротші терміни окремих підсистем меншим числом розробників. Практика показує, що навіть за наявності повністю завершеного проекту, впровадження йде послідовно по окремих підсистемах;

– 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 (систем управління базами даних (СУБД), операційних систем, мов і систем програмування);

– технологія повинна бути підтримана комплексом узгоджених CASE-засобів, що забезпечують автоматизацію процесів, що виконуються на всіх стадіях ЖЦ. Загальний підхід до оцінки і вибору CASE-засобів описаний в розділі 4, приклади комплексів CASE-засобів - в підрозділі 5.7.

Реальне застосування будь-якої технології проектування, розробки і супроводу ІС в конкретній організації і конкретному проекті неможливо без вироблення ряду стандартів (правил, угод), які повинні дотримуватися всіма учасниками проекту. До таких стандартів відносяться наступні:

- стандарт проектування;
- стандарт оформлення проектної документації;
- стандарт призначеного для користувача інтерфейсу.

Стандарт проектування повинен встановлювати:

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

– правила фіксації проектних рішень на діаграмах, зокрема: правила іменування об'єктів (включаючи угоди по термінології), набір атрибутів для всіх об'єктів і правила їх заповнення на кожній стадії, правила оформлення діаграм, включаючи вимоги до форми і розмірів об'єктів, і т. д.;

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

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

Стандарт оформлення проектної документації повинен встановлювати:

- комплектність, склад і структуру документації на кожній стадії

проектування;

- вимоги до її оформлення (включаючи вимоги до змісту розділів, підрозділів, пунктів, таблиць і так далі)
- правила підготовки, розгляду, узгодження і затвердження документації з вказівкою 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 колірна палітра), склад і розташування вікон і елементів управління;
- правила використання клавіатури і миші;
- правила оформлення текстів допомоги;
- перелік стандартних повідомлень;
- правила обробки реакції користувача.

### **1.3.2. Методологія RAD**

Одним з можливих підходів до розробки ПЗ в рамках спіральної моделі ЖЦ є та, що набула останнім часом широкого поширення методологія швидкої розробки застосувань RAD (Rapid Application Development). Під цим терміном зазвичай розуміється процес розробки ПЗ, що містить 3 елементи:

- невелику команду програмістів (від 2 до 10 чоловік);
- короткий, але ретельно такий, що пропрацював виробничий графік (від 2 до 6 міс.);
- цикл, що повторюється, при якому розробники, у міру того, як застосування починає знаходити форму, запрошують і реалізують в продукті вимоги, отримані через взаємодію із замовником.

Команда розробників повинна вдавати із себе групу професіоналів, що мають досвід в аналізі, проектуванні, генерації коду і тестуванні ПЗ з використанням CASE-засобів. Члени колективу повинні також уміти трансформувати в робочі прототипи пропозиції кінцевих користувачів.

Життєвий цикл ПЗ за методологією RAD складається з чотирьох фаз:

- фаза аналізу і планування вимог;
- фаза проектування;
- фаза побудови;
- фаза впровадження.

На фазі аналізу і планування вимог користувачі системи визначають функції, які вона повинна виконувати, виділяють найбільш пріоритетні з них, такі, що вимагають опрацювання в першу чергу, описують інформаційні потреби. Визначення вимог виконується в основному силами користувачів під керівництвом фахівців-розробників. Обмежується масштаб проекту, визначаються тимчасові рамки для кожної з подальших фаз. Крім того, визначається сама можливість реалізації даного проекту у встановлених рамках фінансування, на даних апаратних засобах і тому подібне Результатом

даної фази повинні бути список і пріоритетність функцій майбутньої ІС, попередні функціональні і інформаційні моделі ІС.

На фазі проектування частина користувачів бере участь в технічному проектуванні системи під керівництвом фахівців-розробників. CASE-засоби використовуються для швидкого отримання працюючих прототипів застосувань. Користувачі, безпосередньо взаємодіючи з ними, уточнюють і доповнюють вимоги до системи, які не були виявлені на попередній фазі. Детальніше розглядаються процеси системи. Аналізується і, при необхідності, коректується функціональна модель. Кожен процес розглядається детально. При необхідності для кожного елементарного процесу створюється частковий прототип: екран, діалог, звіт, що знімає неясності або неоднозначності. Визначаються вимоги розмежування доступу до даним. На цій же фазі відбувається визначення набору необхідної документації.

Після детального визначення складу процесів оцінюється кількість функціональних елементів системи, що розробляється, і ухвалюється рішення про розділення ІС на підсистеми, реалізації, що піддаються, однією командою розробників за прийнятний для RAD-проектів час - порядку 60 - 90 днів. З використанням CASE-засобів проект розподіляється між різними командами (ділиться функціональна модель). Результатом даної фази повинні бути:

- загальна інформаційна модель системи;
- функціональні моделі системи в цілому і підсистем, що реалізуються окремими командами розробників;
- точно визначені за допомогою CASE-засобів інтерфейси між підсистемами, що автономно розробляються;
- побудовані прототипи екранів, звітів, діалогів.

Всі моделі і прототипи повинні бути отримані із застосуванням тих CASE-засобів, які використовуватимуться надалі при побудові системи. Дана вимога викликана тим, що в традиційному підході при передачі інформації про проект з етапу на етап може відбутися фактично неконтрольоване спотворення даних. Застосування єдиного середовища зберігання інформації про проект дозволяє уникнути цієї небезпеки.

На відміну від традиційного підходу, при якому використовувалися специфічні засоби прототипування, не призначені для побудови реальних застосувань, а прототипи викидалися після того, як виконували завдання усунення неясностей в проекті, в підході RAD кожен прототип розвивається в частину майбутньої системи. Таким чином, на наступну фазу передається повніша і корисніша інформація.

На фазі побудови виконується безпосередньо сама швидка розробка застосування. На даній фазі розробники проводять ітеративну побудову реальної системи на основі отриманих в попередній фазі моделей, а також вимог нефункціонального характеру. Програмний код частково формується за допомогою автоматичних генераторів, одержуючих інформацію безпосередньо з репозиторія CASE-засобів. Кінцеві користувачі на цій фазі

оцінюють отримувані результати і вносять корективи, якщо в процесі розробки система перестає задовольняти визначеним раніше вимогам. Тестування системи здійснюється безпосередньо в процесі розробки.

Після закінчення робіт кожної окремої команди розробників проводиться поступова інтеграція даної частини системи з останніми, формується повний програмний код, виконується тестування спільної роботи даної частини застосування з останніми, а потім тестування системи в цілому. Завершується фізичне проектування системи:

- визначається необхідність розподілу даних;
- проводиться аналіз використання даних;
- проводиться фізичне проектування бази даних;
- визначаються вимоги до апаратних ресурсів;
- визначаються способи збільшення продуктивності;
- завершується розробка документації проекту.

Результатом фази є готова система, що задовольняє всім узгодженим вимогам.

На фазі впровадження проводиться навчання користувачів, організаційні зміни і паралельно з впровадженням нової системи здійснюється робота з існуючою системою (до повного впровадження нової). Оскільки фаза побудови достатньо нетривала, планування і підготовка до впровадження повинні починатися заздалегідь, як правило, на етапі проектування системи. Приведена схема розробки ІС не є абсолютною. Можливі різні варіанти, залежні, наприклад, від початкових умов, в яких ведеться розробка: розробляється абсолютно нова система; вже було проведено обстеження підприємства і існує модель його діяльності; на підприємстві вже існує деяка ІС, яка може бути використана як початковий прототип або повинна бути інтегрована з тією, що розробляється.

Слідє, проте, відзначити, що методологія RAD, як і будь-яка інша, не може претендувати на універсальність, вона хороша в першу чергу для відносно невеликих проектів, що розробляються для конкретного замовника. Якщо ж розробляється типова система, яка не є закінченим продуктом, а є комплексом типових компонент, централізованих супроводжуваних, таких, що адаптуються до програмно-технічних платформ, СУБД, засобів телекомунікації, організаційно-економічними особливостями об'єктів впровадження і інтегрованих з існуючими розробками, на перший план виступають такі показники проекту, як керованість і якість, які можуть увійти до суперечності з простотою і швидкістю розробки. Для таких проектів необхідні високий рівень планування і жорстка дисципліна проектування, строге проходження заздалегідь розробленими протоколами і інтерфейсами, що знижує швидкість розробки.

Методологія RAD непридатна для побудови складних розрахункових програм, операційних систем або програм управління космічними кораблями, тобто програм, що вимагають написання великого об'єму (сотні тисяч рядків) унікальної коди.

Не підходять для розробки за методологією RAD застосування, в яких відсутня яскраво виражена інтерфейсна частина, що наочно визначає логіку роботи системи (наприклад, застосування реального часу) і застосування, від яких залежить безпека людей (наприклад, керування літаком або атомною електростанцією), оскільки ітеративний підхід припускає, що перші декілька версій напевно не буде повністю працездатні, що в даному випадку виключається.

Оцінка розміру застосувань проводиться на основі так званих функціональних елементів (екрани, повідомлення, звіти, файли і тому подібне) Подібна метрика не залежить від мови програмування, на якому ведеться розробка. Розмір застосування, яке може бути виконане за методологією RAD, для добре відладженого середовища розробки ІС з максимальним повторним використанням програмних компонентів, визначається таким чином:

<b>&lt; 1000</b> функціональних елементів	одна людина
<b>1000-4000</b> функціональних елементів	одна команда розробників
<b>&gt; 4000</b> функціональних елементів	4000 функціональних елементів на одну команду розробників

Як підсумок перерахуємо основні принципи методології RAD:

- розробка застосувань ітераціями;
- необов'язковість повного завершення робіт на кожному з етапів життєвого циклу;
- обов'язкове залучення користувачів до процесу розробки ІС;
- необхідне застосування CASE-засобів, що забезпечують цілісність проекту;
- застосування засобів управління конфігурацією, що полегшують внесення змін до проекту і супроводу готової системи;
- необхідне використання генераторів коду;
- використання прототипування, що дозволяє повніше з'ясувати і задовольнити потреби кінцевого користувача;
- тестування і розвиток проекту, здійснювані одночасно з розробкою;
- ведення розробки нечисленною добре керованою командою професіоналів;
- грамотне керівництво розробкою системи, чітке планування і контроль виконання робіт.

## **2. Структурний підхід до проектування ІС**

### **2.1. Суть структурного підходу**

Суть структурного підходу до розробки ІС полягає в її декомпозиції (розбитті) на функції, що автоматизуються: система розбивається на функціональні підсистеми, які у свою чергу діляться на підфункції, що підрозділяються на завдання і так далі. Процес розбиття продовжується аж до конкретних процедур. При цьому система, що автоматизується, зберігає цілісне уявлення, в якому всі компоненти, що становлять, взаємопов'язані. При розробці системи "знизу-вгору" від окремих завдань до всієї системи цілісність втрачається, виникають проблеми при інформаційній стиковці окремих компонентів.

Всі найбільш поширені методології структурного підходу [9,11,12,13] базуються на ряду загальних принципів [3]. Як два базові принципи використовуються наступні:

- принцип "розділяй і володарюй" - принцип вирішення складних проблем шляхом їх розбиття на безліч менших незалежних завдань, легених для розуміння і рішення;
- принцип ієрархічного впорядкування - принцип організації складових частин проблеми в ієрархічні деревовидні структури з додаванням нових деталей на кожному рівні.

Виділення двох базових принципів не означає, що решта принципів є другорядними, оскільки ігнорування будь-якого з них може привести до непередбачуваних наслідків (у тому числі і до провалу всього проекту). Основними з цих принципів є наступні:

- принцип абстрагування - полягає у виділенні істотних аспектів системи і відвернення від неістотних;
- принцип формалізації - полягає в необхідності строгого методичного підходу до вирішення проблеми;
- принцип несуперечності - полягає в обґрунтованості і узгодженості елементів;
- принцип структуризації даних - полягає в тому, що дані повинні бути структуровані і ієрархічно організовані.

У структурному аналізі використовуються в основному дві групи засобів, що ілюструють функції, що виконуються системою і відносини між даними. Кожній групі засобів відповідають певні види моделей (діаграм), найбільш поширеними серед яких є наступні:

- SADT (Structured Analysis and Design Technique) моделі і відповідні функціональні діаграми (підрозділ 2.2);
- DFD (Data Flow Diagrams) діаграми потоків даних (підрозділ 2.3);
- ERD (Entity-Relationship Diagrams) діаграми "суть-зв'язок" (підрозділ 2.4).

На стадії проектування ІС моделі розширюються, уточнюються і доповнюються діаграмами, що відображають структуру програмного забезпечення: архітектуру ПЗ, структурні схеми програм і діаграми екранних

форм.

Перераховані моделі в сукупності дають повний опис ІС незалежно від того, чи є вона такою, що існує або знов розробляється. Склад діаграм у кожному конкретному випадку залежить від необхідної повноти опису системи.

## **2.2. Методологія функціонального моделювання SADT**

Методологія SADT розроблена Дугласом Россом і отримала подальший розвиток в роботі [4]. На її основі розроблена, зокрема, відома методологія IDEF0 (Icam DEFinition), яка є основною частиною програми ICAM (Інтеграція комп'ютерних і промислових технологій), що проводиться за ініціативою BBC США.

Методологія SADT є сукупністю методів, правил і процедур, призначених для побудови функціональної моделі об'єкту якої-небудь наочної області. Функціональна модель SADT відображає функціональну структуру об'єкту, тобто вироблювані ним дії і зв'язки між цими діями. Основні елементи цієї методології ґрунтуються на наступних концепціях:

- графічне представлення блокового моделювання. Графіка блоків і дуг SADT-діаграмми відображає функцію у вигляді блоку, а інтерфейси входу/виходу представляються дугами, що відповідно входять в блок і виходять з нього. Взаємодія блоків один з одним описуються за допомогою інтерфейсних дуг, що виражають "обмеження", які у свою чергу визначають, коли і яким чином функції виконуються і управляються;

- строгість і точність. Виконання правил SADT вимагає достатньої строгості і точності, не накладаючи в той же час надмірних обмежень на дії аналітика. Правила SADT включають:

- обмеження кількості блоків на кожному рівні декомпозиції (правило 3-6 блоків);

- зв'язність діаграм (номери блоків);

- унікальність влучний і найменувань (відсутність імен, що повторюються);

- синтаксичні правила для графіки (блоків і дуг);

- розділення входів і управлінь (правило визначення ролі даних).

- відділення організації від функції, тобто виключення впливу організаційної структури на функціональну модель.

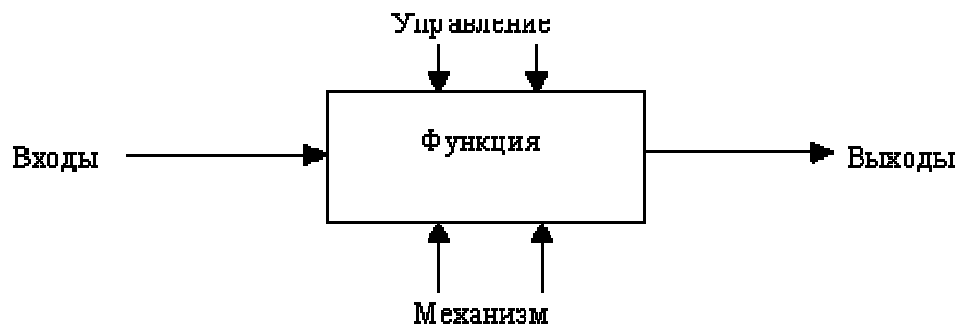
Методологія SADT може використовуватися для моделювання широкого круга систем і визначення вимог і функцій, а потім для розробки системи, яка задовольняє цим вимогам і реалізує ці функції. Для вже існуючих систем SADT може бути використана для аналізу функцій, що виконуються системою, а також для вказівки механізмів, за допомогою яких вони здійснюються.

### **2.2.1. Склад функціональної моделі**

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 і інтерфейси на

них представлені як блоки і дуги. Місце з'єднання дуги з блоком визначає тип інтерфейсу. Інформація, що управляє, входить в 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, представляється дугою, що входить в блок знизу (малюнок 2.1).

Однією з найбільш важливих особливостей методології SADT є поступове введення все великих рівнів деталізації у міру створення діаграм, що відображають модель.



Мал. 2.1. Функціональний блок і інтерфейсні дуги

На малюнку 2.2, де приведено чотири діаграми і їх взаємозв'язки, показана структура SADT-моделі. Кожен компонент моделі може бути декомпозований на іншій діаграмі. Кожна діаграма ілюструє "внутрішню будову" блоку на батьківській діаграмі.

### 2.2.2. Ієрархія діаграм

Побудова SADT-моделі починається з представлення всієї системи у вигляді простої компоненти - одного блоку і дуг, що зображають інтерфейси з функціями поза системою. Оскільки єдиний блок представляє всю систему як єдине ціле, ім'я, вказане в блоці, є загальним. Це вірно і для інтерфейсних дуг - вони також представляють повний набір зовнішніх інтерфейсів системи в цілому.

Потім блок, який представляє систему як єдиний модуль, деталізує на іншій діаграмі за допомогою декількох блоків, сполучених інтерфейсними дугами. Ці блоки представляють 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 уявлення.

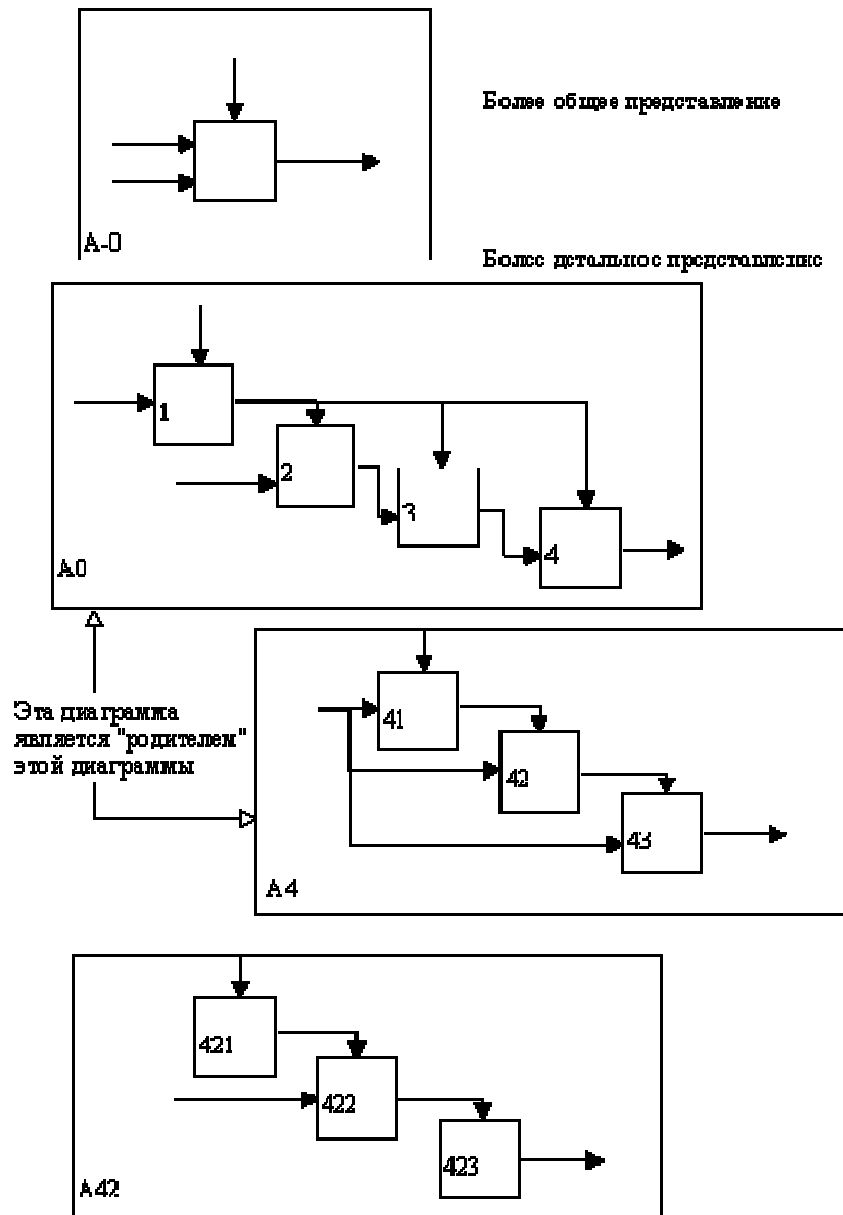
У всіх випадках кожна підфункція може містити тільки ті елементи, які входять в початкову функцію. Крім того, модель не може опустити які-небудь елементи, тобто, як вже наголошувалося, батьківський блок і його інтерфейси забезпечують контекст. До нього не можна нічого додати, і з нього не може бути нічого видалено.

Модель SADT є серією діаграм з супровідною документацією, що розбивають складний об'єкт на складові частини, які представлені 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



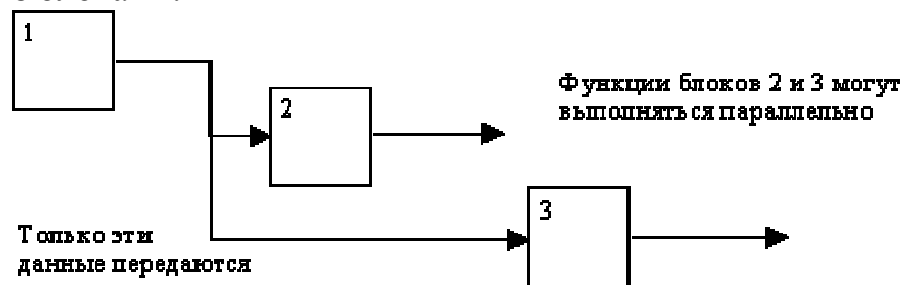
називається батьківською для детальнішої діаграми.

Дуги, що входять в блок і виходять з нього на діаграмі верхнього рівня, є точно тими ж самими, що і дуги, що входять в діаграму нижнього рівня і виходять з неї, тому що блок і діаграма представляють одну і ту ж частину системи.

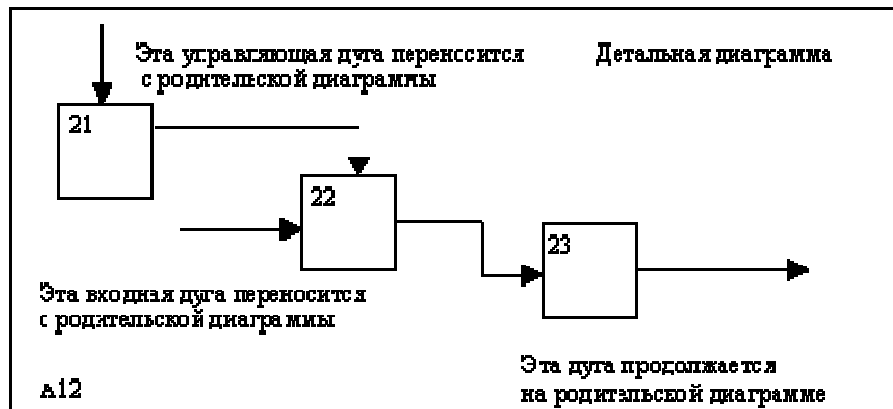
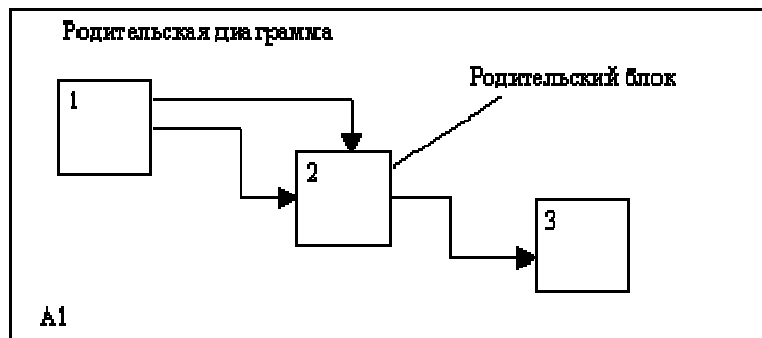


Мал. 2.2. Структура SADT-модели. Декомпозиция диаграм

На картинках 2.3 - 2.5 представлены различные варианты выполнения функций и соединения дуг с блоками.



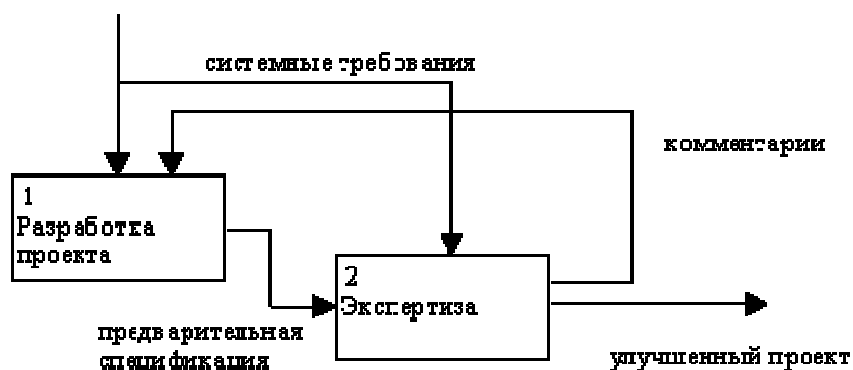
Мал. 2.3. Одновременное выполнение



Мал. 2.4. Відповідність повинна бути повною і несуперечливою

Деякі дуги приєднані до блоків діаграми обома кінцями, у інших же один кінець залишається неприєднаним. Неприєднані дуги відповідають входам, управлінням і виходам батьківського блоку. Джерело або одержувач цих прикордонних дуг може бути виявлений тільки на батьківській діаграмі. Неприєднані кінці повинні відповідати дугам на початковій діаграмі. Всі граничні дуги повинні продовжуватися на батьківській діаграмі, щоб вона була повною і несуперечливою.

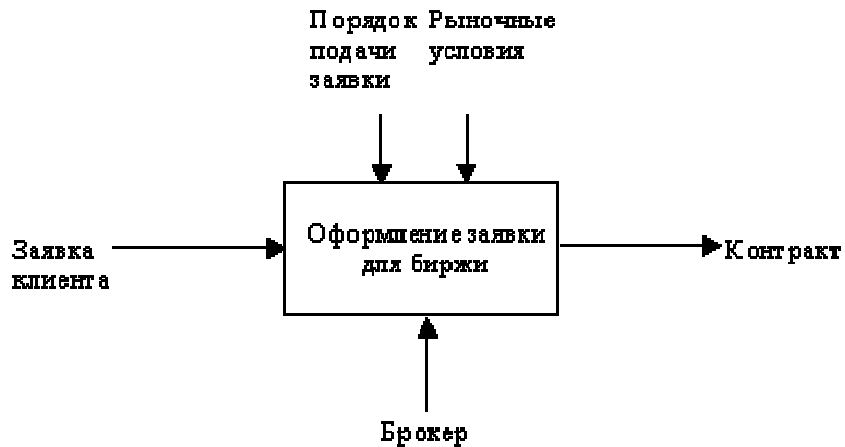
На SADT-діаграмах не вказані явно ні послідовність, ні час. Зворотні зв'язки, ітерації, процеси, що продовжуються, і функції, що перекриваються (за часом), можуть бути зображені за допомогою дуг. Зворотні зв'язки можуть виступати у вигляді коментарів, зауважень, виправлень і так далі (малюнок 2.5).



Мал. 2.5. Приклад зворотного зв'язку

Як було відмічено, 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

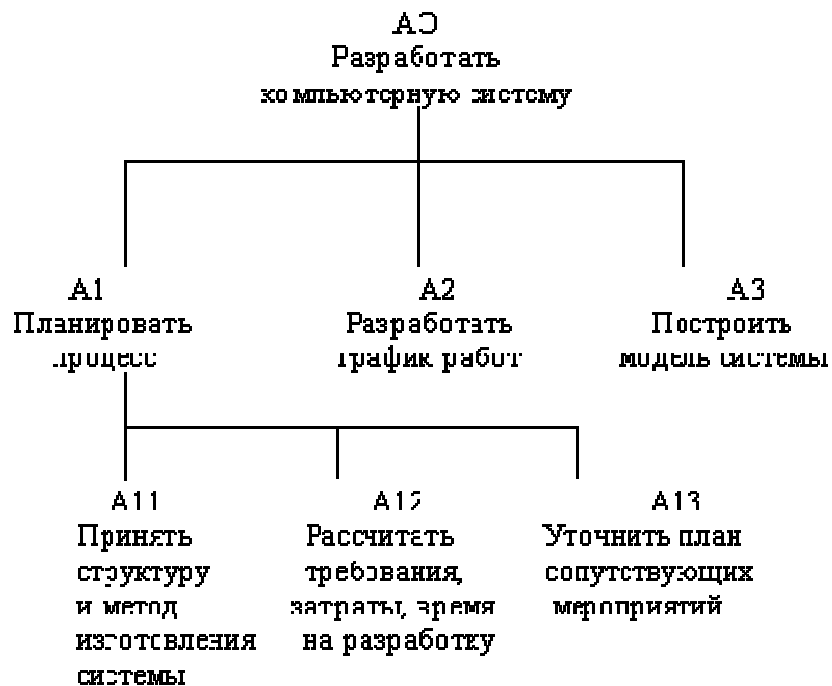
допомагає виконувати дану функцію (малюнок 2.6).



Мал. 2.6. Приклад механізму

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 необхідного числа діаграм. Таким чином, формується ієрархія діаграм.

Для того, щоб вказати положення будь-якої діаграми або блоку в ієрархії, використовуються номери діаграм. Наприклад, A21 є діаграмою, яка деталізує блок 1 на діаграмі A2. Аналогічно, A2 деталізує блок 2 на діаграмі A0, яка є самою верхньою діаграмою моделі. На малюнку 2.7 показано типове дерево діаграм.



Мал. 2.7. Ієрархія діаграм

### 2.2.3. Типи зв'язків між функціями

Одним з важливих моментів при проектуванні ІС за допомогою методології SADT є точна узгодженість типів зв'язків між функціями. Розрізняють принаймні сім типів скріплення:

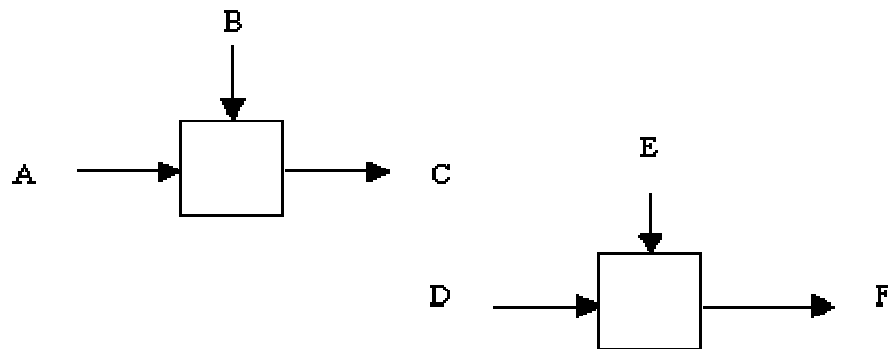
Тип зв'язку	Відносна значущість
-------------	---------------------

Випадкова	0
Логічна	1
Тимчасова	2
Процедурна	3
Комунікаційна	4
Послідовна	5
Функціональна	6

Нижче кожен тип зв'язку стисло визначений і проілюстрований за допомогою типового прикладу з SADT.

**(0) Тип випадкової зв'язності:** найменш бажаний.

Випадкова зв'язність виникає, коли конкретний зв'язок між функціями малий або повністю відсутній. Це відноситься до ситуації, коли імена даних на SADT-дугах в одній діаграмі мають малий зв'язок один з одним. Крайній варіант цього випадку показаний на малюнку 2.8.

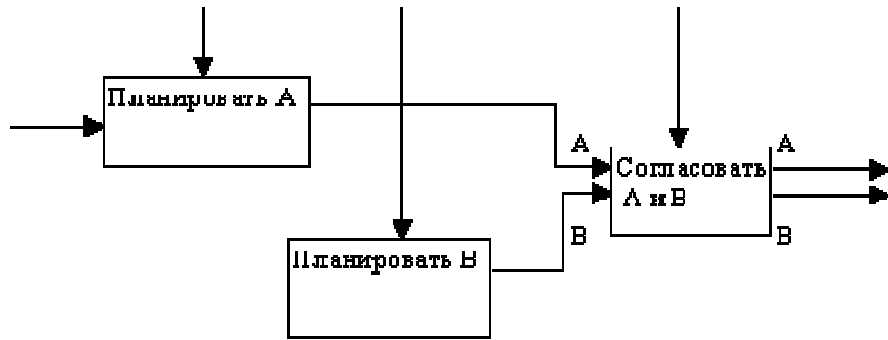


Мал. 2.8. Випадкова зв'язність

**(1) Тип логічної зв'язності.** Логічне скріплення відбувається тоді, коли дані і функції збираються разом унаслідок того, що вони потрапляють в загальний клас або набір елементів, але необхідних функціональних відносин між ними не виявляється.

**(2) Тип тимчасової зв'язності.** Зв'язані за часом елементи виникають унаслідок того, що вони представляють функції, зв'язані в часі, коли дані використовуються одночасно або функції включаються паралельно, а не послідовно.

**(3) Тип процедурної зв'язності.** Процедурно-зв'язані елементи з'являються згрупованими разом унаслідок того, що вони виконуються протягом однієї і тієї ж частини циклу або процесу. Приклад процедурно-зв'язаної діаграми приведений на малюнку 2.9.

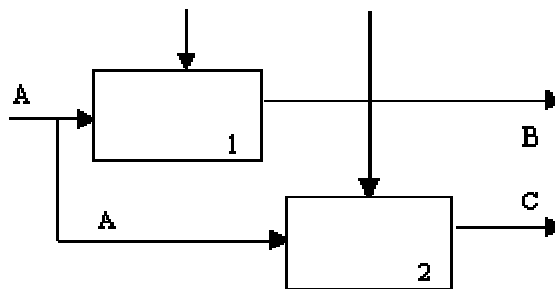


Мал. 2.9. Процедурна зв'язність

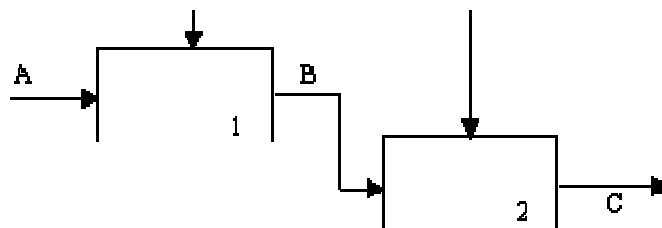
(4) **Тип комунікаційної зв'язності.** Діаграми демонструють комунікаційні зв'язки, коли блоки групуються унаслідок того, що вони використовують одні і ті ж вхідні дані і/або проводять одні і ті ж вихідні дані (малюнок 2.10).

(5) **Тип послідовної зв'язності.** На діаграмах, що мають послідовні зв'язки, вихід однієї функції служить вхідними даними для наступної функції. Зв'язок між елементами на діаграмі є тіснішим, ніж на розглянутих вище рівнях зв'язок, оскільки моделюються причинно-наслідкові залежності (малюнок 2.11).

(6) **Тип функціональної зв'язності.** Діаграма відображає повну функціональну зв'язність, за наявності повної залежності однієї функції від іншої. Діаграма, яка є чисто функціональною, не містить чужорідних елементів, що відносяться до послідовного або слабкішого типу зв'язності. Одним із способів визначення функціонально-зв'язаних діаграм є розгляд двох блоків, зв'язаних через дуги, що управляють, як показано на малюнку 2.12.



Мал. 2.10. Комунікаційна зв'язність

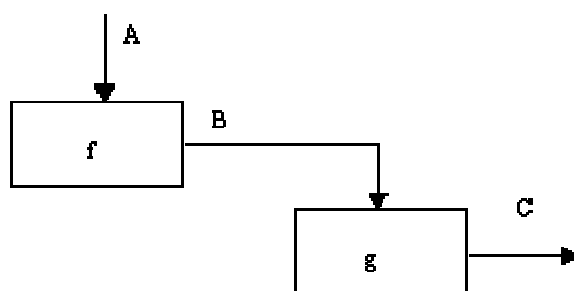


Мал. 2.11. Послідовна зв'язність

У математичних термінах необхідна умова для простого типу функціональної зв'язності, показаної на малюнку 2.12, має наступний вигляд:

$$C = g(B) = g(f(A))$$

Нижче в таблиці представлені всі типи зв'язків, розглянуті вище. Важливо відзначити, що рівні 4-6 встановлюють типи зв'язностей, які розробники вважають найважливішими для отримання діаграм хорошої якості.



Мал. 2.12. Функціональна зв'язність

Значущість	Тип зв'язності	Для функцій	Для даних
0	Випадкова	Випадкова	Випадкова
1	Логічна	Функції однієї і тієї ж множини або типу (наприклад, "редагувати всі входи")	Дані однієї і тієї ж множини або типу
2	Тимчасова	Функції одного і того ж періоду часу (наприклад, "операції ініціалізації")	Дані, використовувані в якому-небудь тимчасовому інтервалі
3	Процедурна	Функції, що працюють в одній і тій же фазі або ітерації (наприклад, "перший прохід компілятора")	Дані, використовувані під час однієї і тієї ж фази або ітерації
4	Коммуникаційна	Функції, що використовують одні і ті ж дані	Дані, на які впливає одна і та ж діяльність
5	Послідовна	Функції, що виконують послідовні перетворення одних і тих же даних	Дані, що перетворюються послідовними функціями
6	Функціональна	Функції, що об'єднуються для виконання однієї функції	Дані, пов'язані з однією функцією

## 2.3. Моделювання потоків даних (процесів)

У основі даної методології (методології Gane/Sarson [11]) лежить побудова моделі аналізованої ІС - проектованою або такою, що реально існує. Відповідно до методології модель системи визначається як ієрархія діаграм потоків даних (ДПД або DFD), 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 деталізують за допомогою діаграм нижнього рівня. Така декомпозиція продовжується, створюючи багаторівневу ієрархію діаграм, до тих пір, поки не буде досягнутий такий рівень декомпозиції, на якому процес стають елементарними і деталізувати їх далі неможливо.

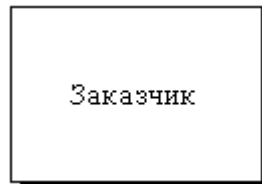
Джерела інформації (зовнішня суть) породжують інформаційні потоки (потоки даних), що переносять інформацію до підсистем або процесів. Ті у свою чергу перетворюють інформацію і породжують нові потоки, які переносять інформацію до інших процесів або підсистем, накопичувачів даних або зовнішньої суті - споживачів інформації. Таким чином, основними компонентами діаграм потоків даних є:

- зовнішня суть;
- системи/підсистеми;
- процеси;
- накопичувачі даних;
- потоки даних.

### 2.3.1. Зовнішня суть

Зовнішня суть є матеріальним предметом або фізичною особою, що є джерелом або приймачем інформації, наприклад, замовники, персонал, постачальники, клієнти, склад. Визначення деякого об'єкту або системи як зовнішня суть указує на те, що вона знаходиться за межами меж аналізованої ІС. В процесі аналізу деяка зовнішня суть може бути перенесені всередину діаграми аналізованої ІС, якщо це необхідно, або, навпаки, частина процесів ІС може бути винесена за межі діаграми і представлена як зовнішня суть.

Зовнішня суть позначається квадратом (малюнок 2.13), розташованим як би "над" діаграмою і що кидає на неї тінь, для того, щоб можна було виділити цей символ серед інших позначень:



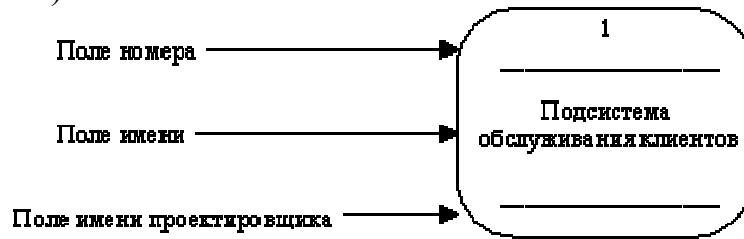
Мал. 2.13. Зовнішня суть

### 2.3.2. Системи і підсистеми

При побудові моделі складною ІС вона може бути представлена в найзагальнішому вигляді на так званій контекстній діаграмі у вигляді однієї системи як єдиного цілого, або може бути декомпозована на ряд підсистем.

Підсистема (або система) на контекстній діаграмі зображається таким

чином (малюнок 2.14).



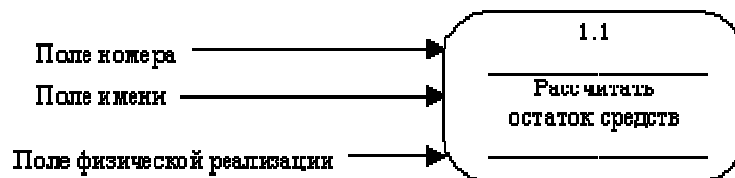
Мал. 2.14. Підсистема

Номер підсистеми служить для її ідентифікації. У поле імені вводиться найменування підсистеми у вигляді пропозиції з підметом і відповідними визначеннями і доповненнями.

### 2.3.3. Процеси

Процесом є перетворення вхідних потоків даних у вихідні відповідно до певного алгоритму. Фізично процес може бути реалізований різними способами: це може бути підрозділ організації (відділ), що виконує обробку вхідних документів і випуск звітів, програма, апаратне реалізоване логічний пристрій і так далі

Процес на діаграмі потоків даних зображається, як показано на малюнку 2.15.



Мал. 2.15. Процес

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

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

Використання таких дієслів, як "обробити", "модернізувати" або "відредагувати" означає, як правило, недостатньо глибоке розуміння даного процесу і вимагає подальшого аналізу.

Інформація в полі фізичної реалізації показує, який підрозділ організації, програма або апаратний пристрій виконує даний процес.

### 2.3.4. Накопичувачі даних

Накопичувачем даних є абстрактний пристрій для зберігання інформації, яку можна у будь-який момент помістити в накопичувач і через деякий час витягувати, причому способи приміщення і витягання можуть бути будь-якими.

Накопичувач даних може бути реалізований фізично у вигляді мікрофіші, ящика в картотечі, таблиці в оперативній пам'яті, файлу на магнітному носіїві і так далі. Накопичувач даних на діаграмі потоків даних



зображається, як показано на малюнку 2.16.

D1	Получаемые счета
----	------------------

Мал. 2.16. Накопичувач даних

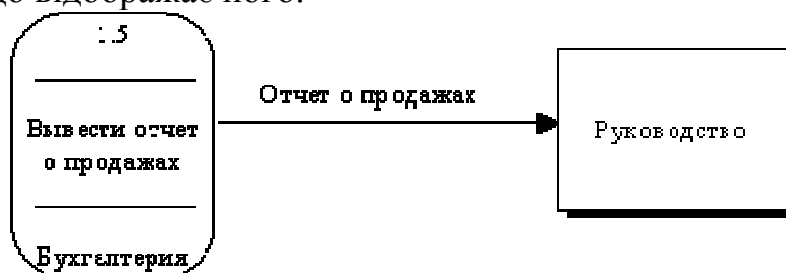
Накопичувач даних ідентифікується буквою "D" і довільним числом. Ім'я накопичувача вибирається з міркування найбільшої інформативності для проектувальника.

Накопичувач даних в загальному випадку є прообразом майбутньої бази даних і опис тих, що зберігаються в ній даних повинно бути пов'язано з інформаційною моделлю.

### 2.3.5. Потіки даних

Потік даних визначає інформацію, передавану через деяке з'єднання від джерела до приймача. Реальний потік даних може бути інформацією, передаваною по кабелю між двома пристроями, листами, що пересилаються поштою, магнітними стрічками або дискетами, переносимими з одного комп'ютера на іншій і так далі

Потік даних на діаграмі зображається лінією, стрілкою, що закінчується, яка показує напрям потоку (малюнок 2.17). Кожен потік даних має ім'я, зміст, що відображає його.



Мал. 2.17. Потік даних

### 2.3.6. Побудова ієрархії діаграм потоків даних

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

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

- наявність великої кількості зовнішньої суті (десять і більш);
- розподілена природа системи;
- багатofункціональність системи з угрупованням функцій, що вже склалося або виявленою, в окремі підсистеми.

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

наступного рівня деталізують контекст і структуру підсистем.

Ієрархія контекстних діаграм визначає взаємодію основних функціональних підсистем проекрованою 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.

Розробка контекстних діаграм вирішує проблему строгого визначення функціональної структури ІС на найранішій стадії її проектування, що особливо важливе для складних багатофункціональних систем, в 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 і ізолюваність об'єктів (відсутність інформаційних зв'язків з іншими об'єктами).

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

- правило балансування - означає, що при деталізації підсистеми або процесу деталізуюча діаграма як зовнішні джерела/приймачі даних може мати тільки ті компоненти (підсистеми, процеси, зовнішня суть, накопичувачі даних), з якими має інформаційний зв'язок підсистема, що деталізується, або процес на батьківській діаграмі;

- правило нумерації - означає, що при деталізації процесів повинна підтримуватися їх ієрархічна нумерація. Наприклад, процеси, що деталізують процес з номером 12, отримують номери 12.1, 12.2, 12.3 і так далі

Мініспецифікація (опис логіки процесу) повинна формулювати його основні функції так, щоб надалі фахівець, що виконує реалізацію проекту, зміг виконати їх або розробити відповідну програму.

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

- наявність у процесу відносно невеликої кількості вхідних і вихідних потоків даних (2-3 потоки);

- можливості опису перетворення даних процесом у вигляді послідовного алгоритму;

- виконання процесом єдиної логічної функції перетворення вхідної інформації у вихідну;

- можливості опису логіки процесу за допомогою мініспецифікації невеликого об'єму (не більше 20-30 рядків).

При побудові ієрархії ДПД переходити до деталізації процесів слід тільки після визначення змісту всіх потоків і накопичувачів даних, яке описується за допомогою структур даних. Структури даних конструюються з елементів даних і можуть містити альтернативи, умовні входження і ітерації. Умовне входження означає, що даний компонент може бути відсутнім в структурі. Альтернатива означає, що в структуру може входити один з

перерахованих елементів. Ітерація означає входження будь-якого числа елементів у вказаному діапазоні. Для кожного елементу даних може указуватися його тип (безперервні або дискретні дані). Для безперервних даних може указуватися одиниця вимірювання (кг, см і тому подібне), діапазон значень, точність уявлення і форма фізичного кодування. Для дискретних даних може указуватися таблиця допустимих значень.

Після побудови закінченої моделі системи її необхідно верифицировать (перевірити на повноту і узгодженість). У повній моделі всі її об'єкти (підсистеми, процеси, потоки даних) повинні бути детально описані і деталізовані. Виявлені недеталізовані об'єкти слід деталізувати, повернувшись на попередні кроки розробки. У узгодженій моделі для всіх потоків даних і накопичувачів даних повинне виконуватися правило збереження інформації: що всі поступають куди-небудь дані повинні бути лічені, а всі прочитувані дані повинні бути записані.

## **2.4. Моделювання даних**

### **2.4.1. Case-метод Баркера**

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-яку систему баз даних.

Найбільш поширеним засобом моделювання даних є діаграми "суть-зв'язок" (ERD). З їх допомогою визначаються важливі для наочної області об'єкти (суть), їх властивості (атрибути) і відносини один з одним (зв'язки). ERD безпосередньо використовуються для проектування реляційних баз даних.

Нотація ERD була вперше введена П. Ченом (Chen) і отримала подальший розвиток в роботах Баркера [8]. Метод Баркера висловлюватиметься на прикладі моделювання діяльності компанії по торгівлі автомобілями. Нижче приведені витяги з інтерв'ю, проведеного з персоналом компанії.

Головний менеджер: один з основних обов'язків - зміст автомобільного майна. Він повинен знати, скільки заплачено за машини і які накладні витрати. Володіючи цією інформацією, він може встановити нижню ціну, за яку міг би продати даний екземпляр. Крім того, він несе відповідальність за продавців і йому потрібно знати, хто що продає і скільки машин продав кожен з них.

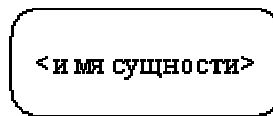
Продавець: йому потрібно знати, яку ціну запрошувати і яка нижня ціна, за яку можна зробити операцію. Крім того, йому потрібна основна інформація про машини: рік випуску, марка, модель і тому подібне

Адміністратор: його завдання зводиться до складання контрактів, для чого потрібна інформація про покупця, автомашину і продавця, оскільки саме контракти приносять продавцям винагороди за продажі.

Перший крок моделювання - витягання інформації з інтерв'ю і виділення суті.

**Суть (Entity)** - реальний або уявний об'єкт, що має істотне значення

для даної наочної області, інформація про яке підлягає зберіганню (малюнок 2.18).



Мал. 2.18. Графічне зображення суті

Кожна суть повинна володіти унікальним ідентифікатором. Кожен екземпляр суті повинен однозначно ідентифікуватися і відрізнятися від всіх інших екземплярів даного типу суті. Кожна суть повинна володіти деякими властивостями:

- кожна суть повинна мати унікальне ім'я, і до одного і того ж імені повинна завжди застосовуватися одна і та ж інтерпретація. Одна і та ж інтерпретація не може застосовуватися до різних імен, якщо тільки вони не є псевдонімами;
- суть володіє одним або декількома атрибутами, які або належать суті, або успадковуються через зв'язок;
- суть володіє одним або декількома атрибутами, які однозначно ідентифікують кожен екземпляр суті;
- кожна суть може володіти будь-якою кількістю зв'язків з іншою суттю моделі.

Звертаючись до приведених вище витягів з інтерв'ю, видно, що суть, яка може бути ідентифіковані з головним менеджером, - це автомашини і продавці. Продавцеві важливі автомашини і пов'язані з їх продажем дані. Для адміністратора важливі покупці, автомашини, продавці і контракти. Виходячи з цього, виділяються 4 суті (автомашина, продавець, покупець, контракт), які зображаються на діаграмі таким чином (малюнок 2.19).



Мал. 2.19.

Наступним кроком моделювання є ідентифікація зв'язків.

**Зв'язок (Relationship)** - поименована асоціація між двома суттю, значуща для даної наочної області. Зв'язок - це асоціація між суттю, при якій, як правило, кожен екземпляр однієї суті, званою батьківською суттю, асоційований з довільною (зокрема нульовим) кількістю екземплярів другої суті, званою суттю-нащадком, а кожен екземпляр суті-нащадка асоційований в точності з одним екземпляром суті-батька. Таким чином, екземпляр суті-нащадка може існувати тільки при існуванні суті батька.

Зв'язку може даватися ім'я, що виражається граматичним оборотом дієслова і що поміщається біля лінії зв'язку. Ім'я кожного зв'язку між двома даною суттю повинне бути унікальним, але імена зв'язків в моделі не зобов'язані бути унікальними. Ім'я зв'язку завжди формується з погляду батька, так що пропозиція може бути утворене з'єднанням імені суті-батька, імені зв'язку, виразу ступеню і імені суті-нащадка.

Наприклад, зв'язок продавця з контрактом може бути виражена таким

чином:

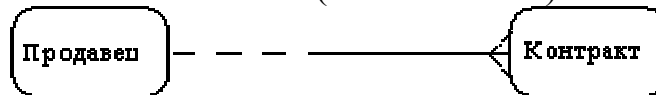
- продавець може отримати винагороду за 1 або більш за контракти;
- контракт повинен бути ініційований рівно одним продавцем.

Ступінь зв'язку і обов'язковість графічно зображаються таким чином (малюнок 2.20).



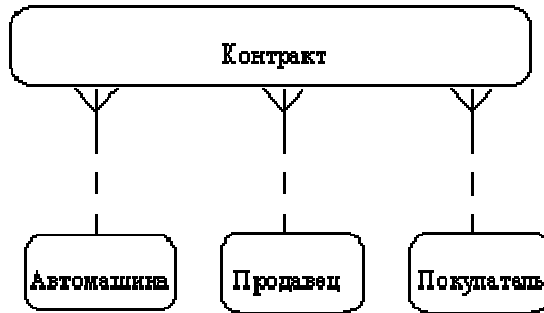
Мал. 2.20.

Таким чином, 2 пропозиції, що описує зв'язок продавця з контрактом, графічно будуть виражена таким чином (малюнок 2.21).



Мал. 2.21.

Описавши також зв'язки решти суті, отримаємо наступну схему (малюнок 2.22).



Мал. 2.22.

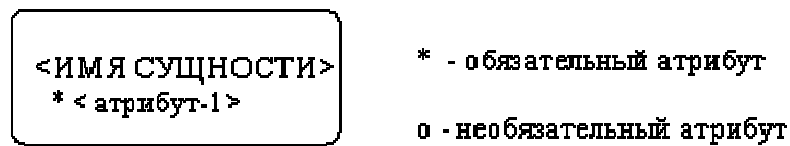
Останнім кроком моделювання є ідентифікація атрибутів.

**Атрибут** - будь-яка характеристика суті, значуща для даної наочної області і призначена для кваліфікації, ідентифікації, класифікації, кількісної характеристики або виразу стану суті. Атрибут представляє тип характеристик або властивостей, що асоціюються з безліччю реальних або абстрактних об'єктів (людей, місць, подій, станів, ідей, пара предметів і так далі). Екземпляр атрибуту - це певна характеристика окремого елемента множини. Екземпляр атрибуту визначається типом характеристики і її значенням, званім значенням атрибуту. У ER-моделі атрибути асоціюються з конкретною суттю. Таким чином, екземпляр суті повинен володіти єдиним певним значенням для асоційованого атрибуту.

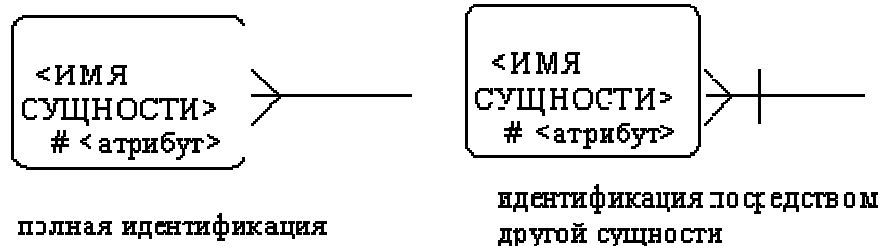
Атрибут може бути або обов'язковим, або необов'язковим (малюнок 2.23). Обов'язковість означає, що атрибут не може приймати невизначених значень (null values). Атрибут може бути або описовим (тобто звичайним дескриптором суті), або входити до складу унікального ідентифікатора (первинного ключа).

**Унікальний ідентифікатор** - це атрибут або сукупність атрибутів і/або зв'язків, призначена для унікальної ідентифікації кожного екземпляра даного типу суті. У разі повної ідентифікації кожен екземпляр даного типу суті повністю ідентифікується своїми власними ключовими атрибутами, інакше в

його ідентифікації беруть участь також атрибути іншої суті-батька (малюнок 2.24).



Мал. 2.23.



Мал. 2.24.

Кожен атрибут ідентифікується унікальним ім'ям, виразимим граматичним оборотом іменника, що описує характеристику, що представляється атрибутом. Атрибути зображаються у вигляді списку імен усередині блоку асоційованої суті, причому кожен атрибут займає окремий рядок. Атрибути, що визначають первинний ключ, розміщуються нагорі списку і виділяються знаком "#".

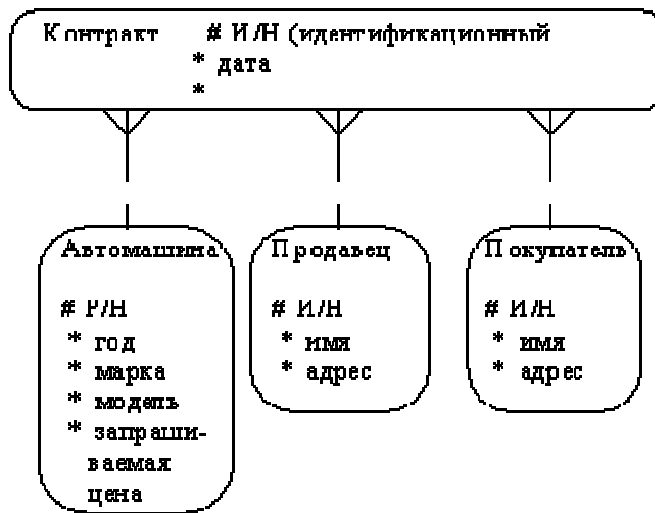
Кожна суть повинна володіти хоч би одним можливим ключем. Можливий ключ суті - це один або декілька атрибутів, чиї значення однозначно визначають кожен екземпляр суті. При існуванні декількох можливих ключів один з них позначається як первинний ключ, а останні - як альтернативні ключі.

З урахуванням наявної інформації доповнимо побудовану раніше діаграму (малюнок 2.25).

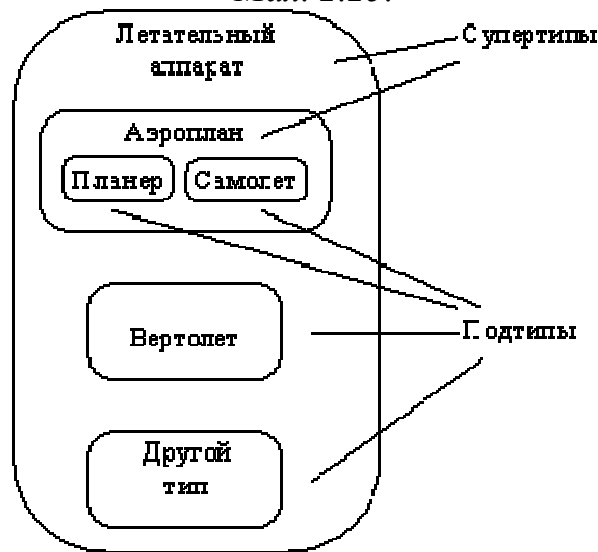
Крім перерахованих основних конструкцій модель даних може містити ряд додаткових.

**Підтипи і супертипи:** одна суть є узагальнювальним поняттям для групи подібної суті (малюнок 2.26).

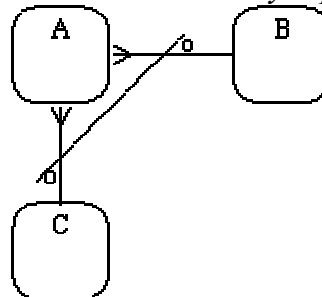
**Зв'язки, що взаємно виключають:** кожен екземпляр суті бере участь тільки в одному зв'язку з групи зв'язків, що взаємно виключають (малюнок 2.27).



Мал. 2.25.



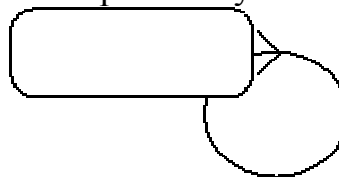
Мал. 2.26. Підтипи і супертипи



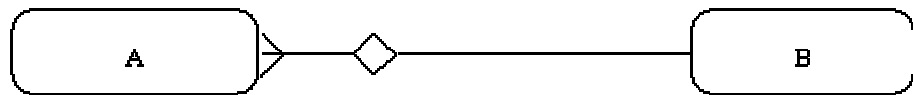
Мал. 2.27. Зв'язки, що взаємно виключають

**Рекурсивний зв'язок:** суті може бути пов'язана сама з собою (малюнок 2.28).

**Непереміщувані (non-transferrable) зв'язки:** екземпляр суті не може бути перенесений з одного екземпляра зв'язку в іншій (малюнок 2.29).



Мал. 2.28. Рекурсивний зв'язок

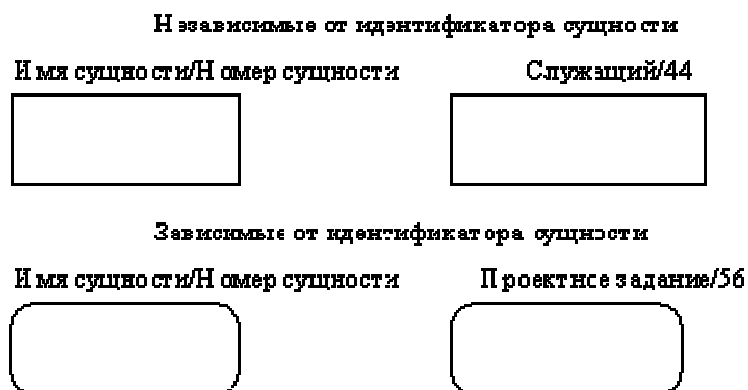


Мал. 2.29. Непереміщений зв'язок

#### 2.4.2. Методологія IDEF1

Метод IDEF1, розроблений Т.Ремей (T.Ramey), також заснований на підході П.Чена і дозволяє побудувати модель даних, еквівалентну реляційній моделі в третій нормальній формі. В даний час 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 поряд поширених CASE-засобів (зокрема, ERwin, Design/IDEF).

Суть в методології IDEF1X є незалежною від ідентифікаторів або просто незалежною, якщо кожен екземпляр суті може бути однозначно ідентифікований без визначення його відносин з іншою суттю. Суть називається залежною від ідентифікаторів або просто залежною, якщо однозначна ідентифікація екземпляра суті залежить від його відношення до іншої суті (малюнок 2.30).



Мал. 2.30. Суть

Кожній суті привласнюється унікальне ім'я і номер, що розділяються косою межею "/" і що поміщаються над блоком.

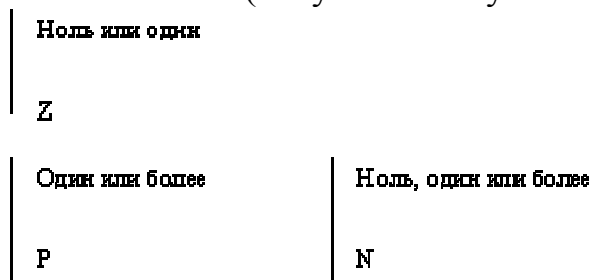
Зв'язок може додатково визначатися за допомогою вказівки ступеню або потужності (кількості екземплярів суті-нащадка, яке може існувати для кожного екземпляра суті-батька). У IDEF1X можуть бути виражені наступні потужності зв'язків:

- кожен екземпляр суті-батька може мати нуль, один або більш пов'язаних з ним екземплярів суті-нащадка;
- кожен екземпляр суті-батька повинен мати не менше одного пов'язаного з ним екземпляра суті-нащадка;
- кожен екземпляр суті-батька повинен мати не більш за один пов'язаний з ним екземпляр суті-нащадка;
- кожен екземпляр суті-батька пов'язаний з деяким фіксованим числом екземплярів суті-нащадка.

Якщо екземпляр суті-нащадка однозначно визначається своїм зв'язком з суттю-батьком, то зв'язок називається такою, що ідентифікує, інакше - що не ідентифікує.

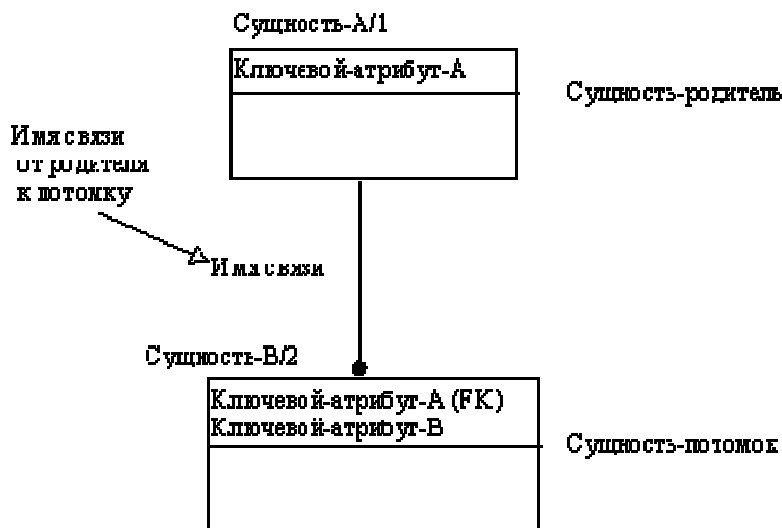


Зв'язок зображається лінією, що проводиться між суттю-батьком і суттю-нащадком з крапкою на кінці лінії у суті-нащадка. Потужність зв'язку позначається як показано на мал. 2.31 (потужність за умовчанням - N).



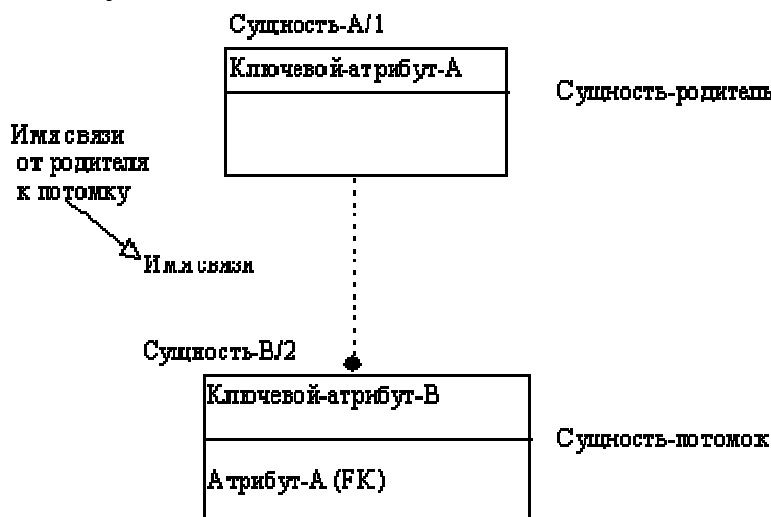
Мал. 2.31. Потужність зв'язку

Ідентифікуючий зв'язок між суттю-батьком і суттю-нащадком зображається суцільною лінією (малюнок 2.32). Суть-нащадок в ідентифікуючому зв'язку є залежною від ідентифікатора суттю. Суть-батько в ідентифікуючому зв'язку може бути як незалежною, так і залежною від ідентифікатора суттю (це визначається її зв'язками з іншою суттю).



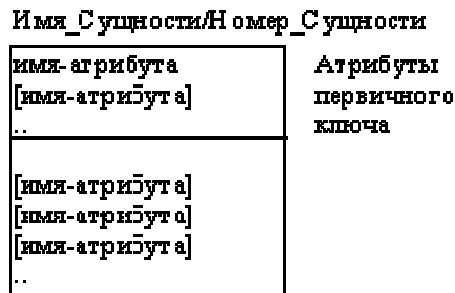
Мал. 2.32. Ідентифікуючий зв'язок

Пунктирна лінія зображає неідентифікуючий зв'язок (малюнок 2.33). Суть-нащадок в неідентифікуючому зв'язку буде незалежною від ідентифікатора, якщо вона не є також суттю-нащадком в якому-небудь ідентифікуючому зв'язку.



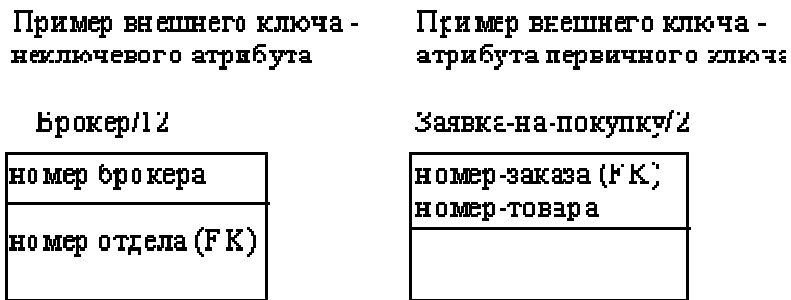
*Мал. 2.33. Неідентифікуючий зв'язок*

Атрибути зображаються у вигляді списку імен усередині блоку суті. Атрибути, що визначають первинний ключ, розміщуються нагорі списку і відділяються від інших атрибутів горизонтальною межею (малюнок 2.34).



*Мал. 2.34. Атрибути і первинні ключі*

Суть може мати також зовнішні ключі (Foreign Key), які можуть використовуватися як частина або цілий первинний ключ або неключовий атрибут. Зовнішній ключ зображається за допомогою приміщення всередину блоку суті імен атрибутів, після яких слідує буква FK в дужках (малюнок 2.35).



*Мал. 2.35. Приклади зовнішніх ключів*

**2.4.3. Підхід, що використовується в CASE-засобі Vantage Team Builder**

У CASE-засобі Vantage Team Builder (Westmount I-CASE) [14] використовується один з варіантів нотації П. Чена. На ER-діаграмах суть позначається прямокутником, що містить ім'я суті (малюнок 2.36), а зв'язок - ромбом, зв'язаним лінією з кожною з взаємодіючої суті. Числа над лініями означають ступінь зв'язку.



*Мал. 2.36. Позначення суті і зв'язків*

Зв'язки є багатонаправленими і можуть мати атрибути (за винятком ключових). Виділяють два види зв'язків:

- необов'язковий зв'язок (optional);
- слабкий зв'язок (weak).

У **необов'язковому зв'язку** (малюнок 2.37) можуть брати участь не всі екземпляри суті.



*Мал. 2.37. Необов'язковий зв'язок*

На відміну від необов'язкового зв'язку в **повному (total)** зв'язку беруть участь всі екземпляри хоч би однієї з суті. Це означає, що екземпляри такого зв'язку існують тільки за умови існування екземплярів іншої суті. Повний зв'язок може мати один з 4-х видів: обов'язковий зв'язок, слабкий зв'язок, зв'язок "супертип-подтип" і асоціативний зв'язок.

**Обов'язковий (mandatory) зв'язок** описує зв'язок між "незалежною" і "залежною" суттю. Всі екземпляри залежної ("обов'язковою") суті можуть існувати тільки за наявності екземплярів незалежної ("необов'язковою") суті, тобто екземпляр "обов'язкової" суті може існувати тільки за умови існування певного екземпляра "необов'язкової" суті.

У прикладі (малюнок 2.38) мається на увазі, що кожен автомобіль має принаймні одного водія, але не кожен службовець управляє машиною.



Мал. 2.38. Обов'язковий зв'язок

У **слабкому зв'язку** існування однієї з суті, що належить деякій множині ("слабкою") залежить від існування певної суті, що належить іншій множині ("сильною"), тобто екземпляр "слабкої" суті може бути ідентифікований тільки за допомогою екземпляра "сильної" суті. Ключ "сильної" суті є частиною складеного ключа "слабкої" суті.

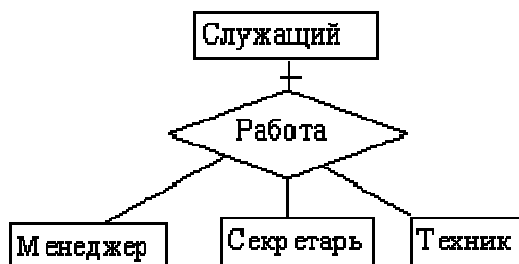
Слабкий зв'язок завжди є бінарним і має на увазі обов'язковий зв'язок для "слабкої" суті. Суть може бути "слабкою" в одному зв'язку і "сильною" в іншій, але не може бути "слабкою" більш, ніж в одному зв'язку. Слабкий зв'язок може не мати атрибутів.

Приклад на малюнку 2.39: ключ (номер) рядка в документі може не бути унікальним і повинен бути доповнений ключем документа.



Мал. 2.39. Слабкий зв'язок

Зв'язок "супертип-подтип" зображена на малюнку 2.40. Загальні характеристики (атрибути) типу визначаються в суті-супертипі, суть-підтип успадковує всі характеристики супертипу. Екземпляр підтипу існує тільки за умови існування певного екземпляра супертипу. Підтип не може мати ключа (він імпортує ключ з супертипу). Суть, що є супертипом в одному зв'язку, може бути підтипом в іншому зв'язку. Зв'язок супертипу не може мати атрибутів.

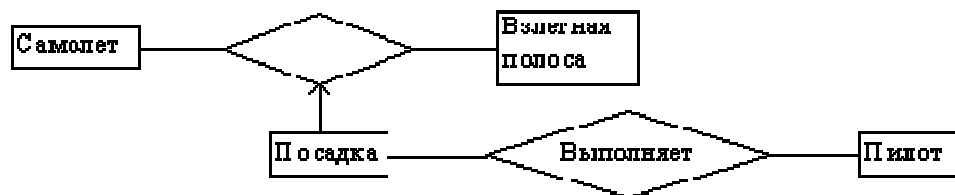


Мал. 2.40. Зв'язок "супертип-подтип"

У **асоціативному зв'язку** кожен екземпляр зв'язку (асоціативний

об'єкт) може існувати тільки за умови існування певних екземплярів кожній з взаємозв'язаної суті. Асоціативний об'єкт - об'єкт, що є одночасно суттю і зв'язком. Асоціативний зв'язок - це зв'язок між декількома "незалежною" суттю і однією "залежною" суттю. Зв'язок між незалежною суттю має атрибути, які визначаються в залежній суті. Таким чином, залежна суть визначається в термінах атрибутів зв'язку між рештою суті.

У прикладі на малюнку 2.41 літак виконує посадку на злітну смугу в заданий час при певній швидкості і напрямі вітру. Оскільки ці характеристики застосовні тільки до конкретної посадки, вони є атрибутами посадки, а не літака або злітної смуги. Пілот, що виконує посадку, пов'язаний набагато сильнішим з конкретною посадкою, чим з літаком або злітною смугою.



Мал. 2.41. Асоціативний зв'язок

Первинний ключ кожного типу суті позначається зірочкою (\*).

ER-діаграма повинна підкорятися наступним правилам:

- кожна суть, кожен атрибут і кожен зв'язок повинні мати ім'я (зв'язок супертипу або асоціативний зв'язок може не мати імені);
- ім'я суті повинне бути унікальне в рамках моделі даних;
- ім'я атрибуту повинне бути унікальне в рамках суті;
- ім'я зв'язку повинне бути унікальне, якщо для неї генерується таблиця БД;
- кожен атрибут повинен мати визначення типу даних;
- суть в необов'язковому зв'язку повинна мати ключовий атрибут. Те ж саме відноситься до сильної суті в слабкому зв'язку, супертипу в зв'язку "супертип-подтип" і необов'язковій суті в обов'язковому (повною) зв'язку;
- підтип в зв'язку "супертип-подтип" не може мати ключового атрибуту;
- у асоціативному або слабкому зв'язку може бути тільки одна асоціативна (слабка) суть;
- зв'язок не може бути одночасне обов'язковою, "супертип-подтип" або асоціативною.

## 2.5. Приклад використання структурного підходу

### 2.5.1. Опис предметної області

У даному прикладі використовується методологія Yourdon [12], реалізована в CASE-засобіве Vantage Team Builder [14].

Як наочна область використовується опис роботи відеобібліотеки, яка отримує запити на фільми від клієнтів і стрічки, повертані клієнтами. Запити розглядаються адміністрацією відеобібліотеки з використанням інформації про клієнтів, фільми і стрічки. При цьому перевіряється і оновлюється список орендованих стрічок, а також перевіряються записи про членство в

бібліотеці. Адміністрація контролює також повернення стрічок, використовуючи інформацію про фільми, стрічках і список орендованих стрічок, який оновлюється. Обробка запитів на фільми і повернень стрічок включає наступні дії: якщо клієнт не є членом бібліотеки, він не має права на оренду. Якщо необхідний фільм є в наявності, адміністрація інформує клієнта про орендну плату. Проте, якщо клієнт прострочив термін повернення стрічок, що є у нього, йому не дозволяється брати нові фільми. Коли стрічка повертається, адміністрація розраховує орендну плату плюс пені за невчасне повернення.

Відеобібліотека отримує нові стрічки від своїх постачальників. Коли нові стрічки поступають в бібліотеку, необхідна інформація про них фіксується. Інформація про членство в бібліотеці міститься окремо від записів про оренду стрічок.

Адміністрація бібліотеки регулярно готує звіти за певний період часу про членів бібліотеки, постачальників стрічок, видачу певних стрічок і стрічки, придбані бібліотекою.

### **2.5.2. Організація проекту**

Весь проект розділяється на 4 фази: аналіз, глобальне проектування (проектування архітектури системи), детальне проектування і реалізація (програмування).

На фазі аналізу будується модель середовища (Environmental Model). Побудова моделі середовища включає:

- аналіз поведінки системи (визначення призначення ІС, побудова початкової контекстної діаграми потоків даних (DFD) і формування матриці списку подій (ELM), побудова контекстних діаграм);
- аналіз даних (визначення складу потоків даних і побудова діаграм структур даних (DSD), конструювання глобальної моделі даних у вигляді ER-діаграми).

Призначення ІС визначає угоду між проектувальниками і замовниками щодо призначення майбутньої ІС, загальний опис ІС для самих проектувальників і межі ІС. Призначення фіксується як текстовий коментар в "нульовому" процесі контекстної діаграми.

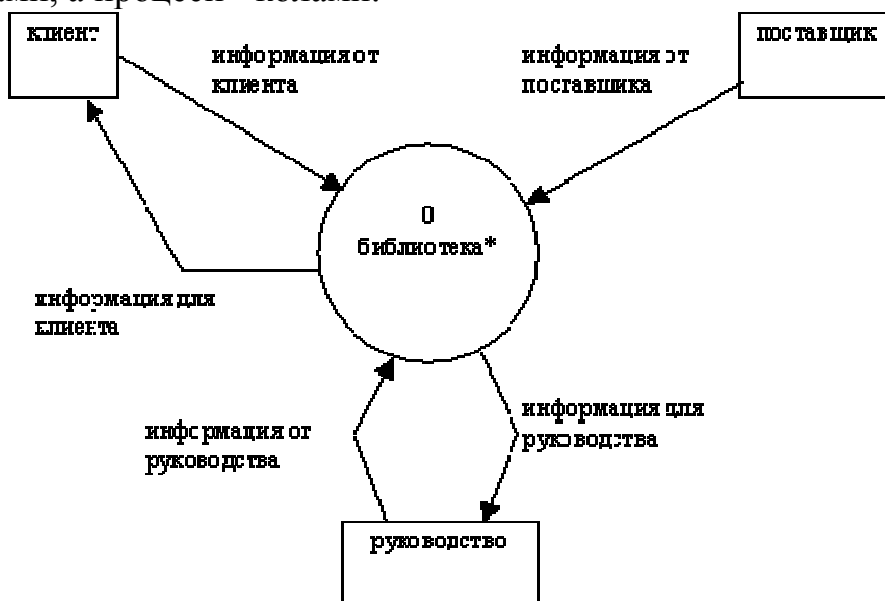
Наприклад, в даному випадку призначення ІС формулюється таким чином: ведення бази даних про членів бібліотеки, фільми, оренду і постачальників. При цьому керівництво бібліотеки повинне мати можливість отримувати різні види звітів для виконання своїх завдань.

Перед побудовою контекстної DFD необхідно проаналізувати зовнішні події (зовнішні об'єкти), що роблять вплив на функціонування бібліотеки. Ці об'єкти взаємодіють з ІС шляхом інформаційного обміну з нею.

З опису наочної області виходить, що в процесі роботи бібліотеки беруть участь наступні групи людей: клієнти, постачальники і керівництво. Ці групи є зовнішніми об'єктами. Вони не тільки взаємодіють з системою, але також визначають її межі і зображаються на початковій контекстною DFD як термінатори (зовнішня суть).

Початкова контекстна діаграма зображена на малюнку 2.42. На відміну

від нотації Gane/Sarson зовнішня суть позначається звичайними прямокутниками, а процеси - колами.



Мал. 2.42. Початкова контекстна діаграма

Список подій будується у вигляді матриці (ELM) і описує різні дії зовнішньої суті і реакцію ІС на них. Цими діями є зовнішні події, що впливають на бібліотеку. Розрізняють наступні типи подій:

Абревіатура	Тип
NC	Нормальне управління
ND	Нормальні дані
NCD	Нормальне управління/данніе
TC	Тимчасове управління
TD	Тимчасові дані
TCD	Тимчасове управління/данніе

Всі дії позначаються як нормальні дані. Ці дані є подіями, які ІС сприймає безпосередньо, наприклад, зміна адреси клієнта, яке повинне бути відразу зареєстроване. Вони з'являються в DFD як вміст потоків даних.

Матриця списку подій має наступний вигляд:

№	Опис	Тип	Реакція
1	Клієнт бажає стати членом бібліотеки	ND	Реєстрація клієнта як член бібліотеки
2	Клієнт повідомляє про зміну адреси	ND	Реєстрація зміненої адреси клієнта
3	Клієнт запрошує оренду фільму	ND	Розгляд запиту
4	Клієнт повертає фільм	ND	Регистрація возврата
5	Руководство надає повноваження новому постачальнику	ND	Регистрація постачальника
6	Постачальник повідомляє про зміну	ND	Регистрація зміненого

	адреса		адреса поставщика
7	Поставщик направляет фильм в библиотеку	ND	Получение нового фильма
8	Руководство запрашивает новый отчет	ND	Формирование требуемого отчета для руководства

Для завершення аналізу функціонального аспекту поведінки системи будується повна контекстна діаграма, що включає діаграму нульового рівня. При цьому процес "бібліотека" декомпозирується на 4 процеси, що відображають основні види адміністративної діяльності бібліотеки. Існуючі "абстрактні" потоки даних між термінаторами і процесами трансформуються в потоки, що представляють обмін даними на конкретнішому рівні. Список подій показує, які потоки існують на цьому рівні: кожна подія із списку повинна формувати деякий потік (подію формує вхідний потік, реакція - вихідний потік). Один "абстрактний" потік може бути роздільний на більш ніж один "конкретний" потік.

<b>Потоки на діаграмі верхнього рівня</b>	<b>Потоки на діаграмі нульового рівня</b>
Інформація від клієнта	Дані про клієнта, Запит про оренду
Інформація для клієнта	Членська картка, Відповідь на запит про оренду
Інформація від керівництва	Запит звіту про нових членів, Новий постачальник, Запит звіту про постачальників, Запит звіту про оренду, Запит звіту про фільми
Інформація для керівництва	Звіт про нових членів, Звіт про постачальників, Звіт про оренду, Звіт про фільми
Інформація від постачальника	Дані про постачальника, Нові фільми

На приведеній DFD (малюнок 2.43) накопичувач даних "бібліотека" є глобальним або абстрактним представленням сховища даних.

Аналіз функціонального аспекту поведінки системи дає уявлення про обмін і перетворення даних в системі. Взаємозв'язок між "абстрактними" потоками даних і "конкретними" потоками даних на діаграмі нульового рівня виражається в діаграмах структур даних (малюнок 2.44).

На фазі аналізу будується глобальна модель даних, "суть-зв'язок", що представляється у вигляді діаграми (малюнок 2.45).

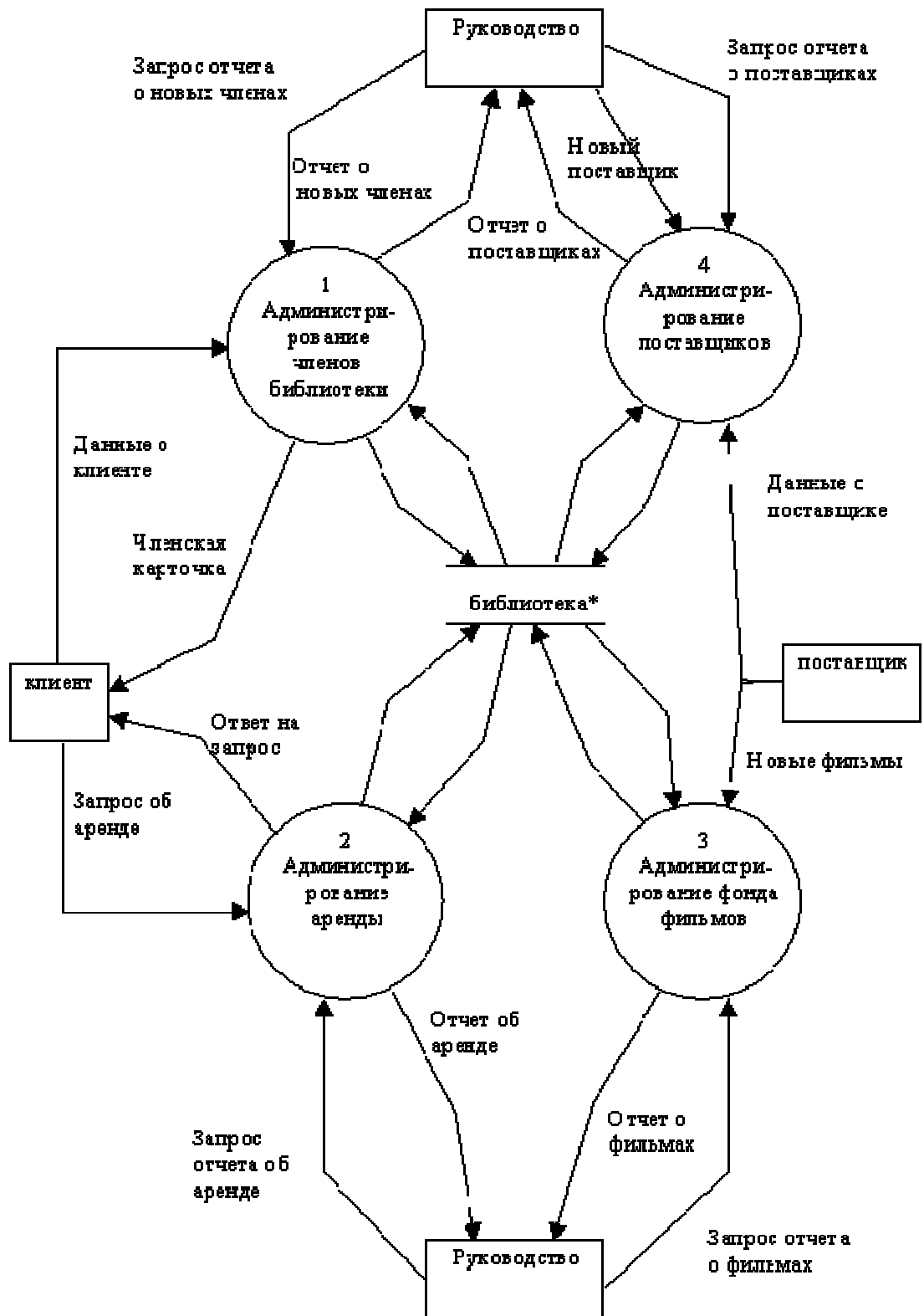
Між різними типами діаграм існують наступні взаємозв'язки:

- ELM-DFD: події - вхідні потоки, реакції - вихідні потоки
- DFD-DSD: потоки даних - структури даних верхнього рівня
- DFD-ERD: накопичувачі даних - ER-діаграми
- DSD-ERD: структури даних нижнього рівня - атрибути суті

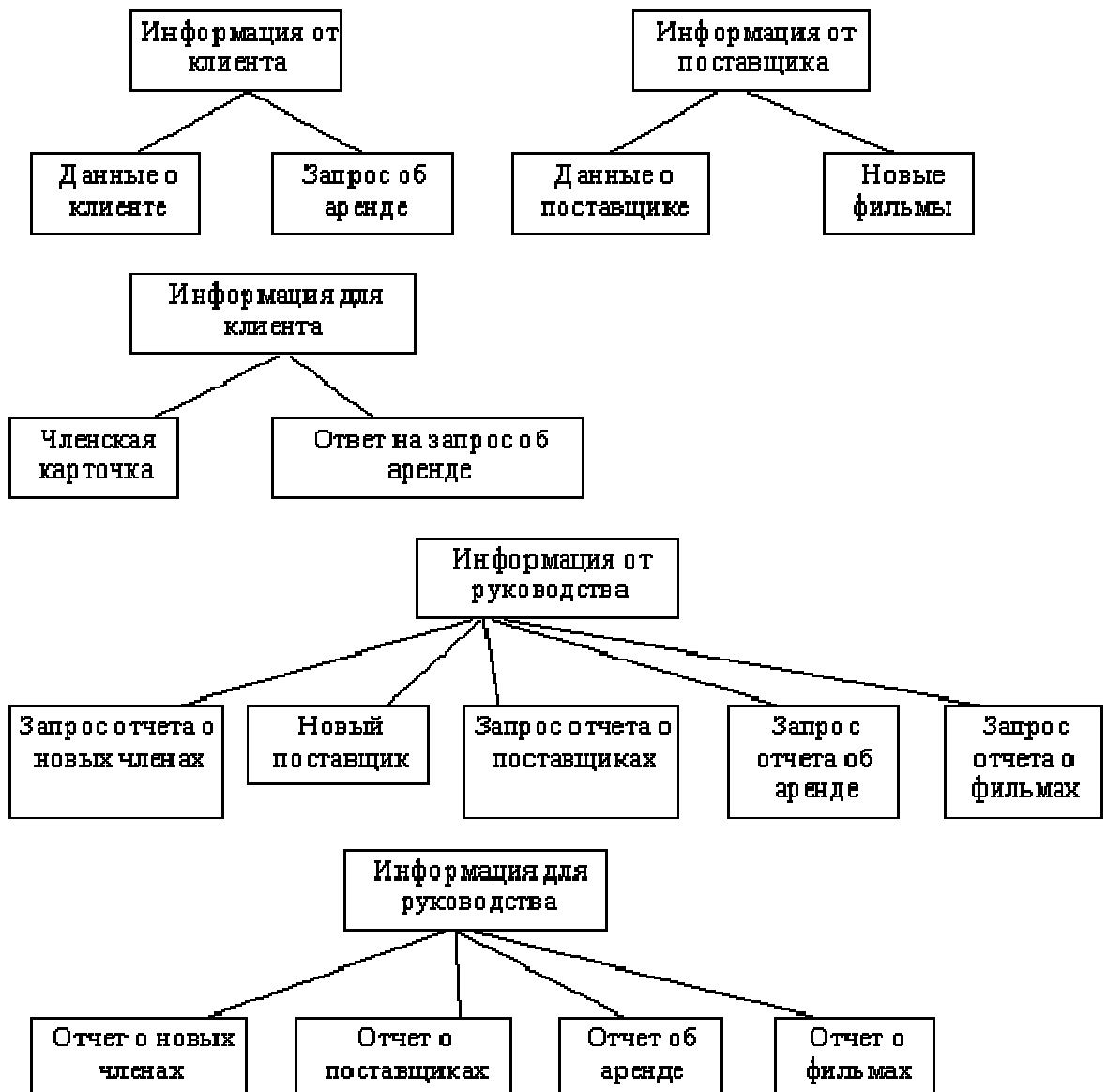
На фазі проектування архітектури будується наочна модель. Процес побудови наочної моделі включає:

- детальний опис функціонування системи;
- подальший аналіз використовуваних даних і побудова логічної моделі даних для подальшого проектування бази даних;
- визначення структури призначеного для користувача інтерфейсу, специфікації форм і порядку їх появи;
- уточнення діаграм потоків даних і списку подій, виділення серед процесів нижнього рівня інтерактивних і неінтерактивних, визначення для них мініспецифікацій.





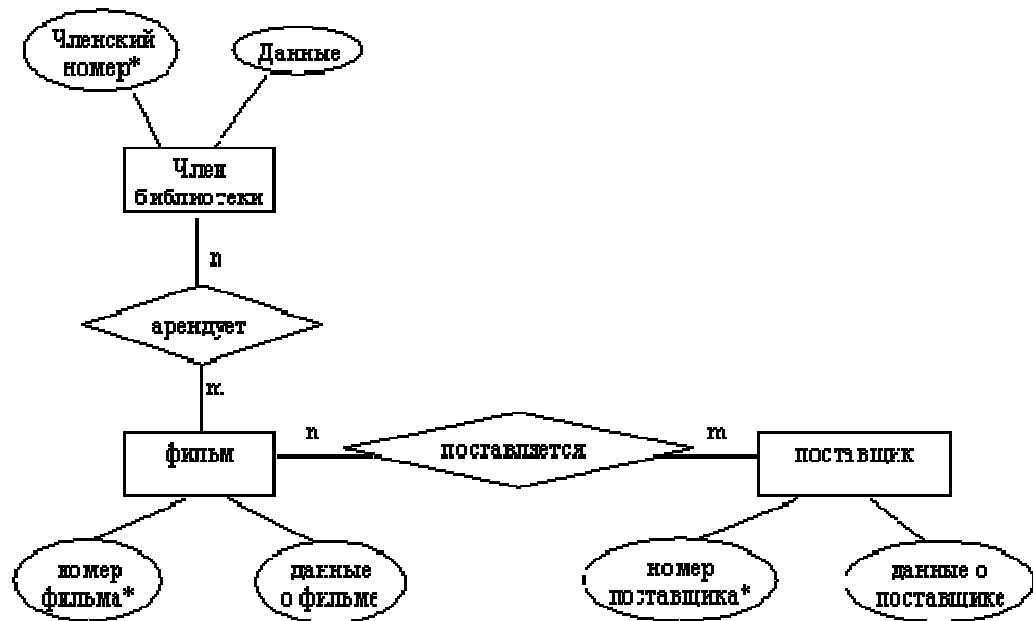
Мал. 2.43. Контекстна діаграма



Мал. 2.44. Диаграмма структур даних

Результатами проектування архітектури є:

- модель процесів (діаграми архітектури системи (SAD) і мініспецифікації на структурованій мові);
- модель даних (ERD і підсхеми ERD);
- модель призначеного для користувача інтерфейсу (класифікація процесів на інтерактивні і неінтерактивні функції, діаграма послідовності форм (FSD - Form Sequence Diagram), що показує, які форми з'являються в застосуванні і в якому порядку. На FSD фіксується набір і структура викликів екранних форм. Діаграми FSD утворюють ієрархію, на вершині якої знаходиться головна форма застосування, що реалізовує підсистему. На другому рівні знаходяться форми, що реалізують процеси нижнього рівня функціональної структури, зафіксованої на діаграмах SAD.



Мал. 2.45. Діаграма "суть-зв'язок"

На фазі детального проектування будується модульна модель. Під модульною моделлю розуміється реальна модель проекрованої прикладної системи. Процес її побудови включає:

- уточнення моделі бази даних для подальшої генерації SQL-предложений;
- уточнення структури призначеного для користувача інтерфейсу;
- побудова структурних схем, що відображають логіку роботи призначеного для користувача інтерфейсу і модель бізнес-логіки (Structure Charts Diagram - SCD) і прив'язка їх до форм.
- Результатами детального проектування є:
- модель процесів (структурні схеми інтерактивних і неінтерактивних функцій);
- модель даних (визначення в ERD всіх необхідних параметрів для застосувань);
- модель призначеного для користувача інтерфейсу (діаграма послідовності форм (FSD), що показує, які форми з'являються в застосуванні і в якому порядку, взаємозв'язок між кожною формою і певною структурною схемою, взаємозв'язок між кожною формою і однією або кількома суттями в ERD).

На фазі реалізації будується реалізаційна модель. Процес її побудови включає:

- генерацію SQL-предложений, що визначають структуру цільової БД (таблиці, індекси, обмеження цілісності);
- уточнення структурних схем (SCD) і діаграм послідовності форм (FSD) з подальшою генерацією коду застосувань.

На основі аналізу потоків даних і взаємодії процесів з сховищами даних здійснюється остаточне виділення підсистем (попереднє повинне було бути зроблене і зафіксоване на етапі формулювання вимог в технічному завданні). При виділенні підсистем необхідно керуватися принципом функціональної зв'язаності і принципом мінімізації інформаційної

залежності. Необхідно враховувати, що на підставі таких елементів підсистеми як процеси і дані на етапі розробки повинне бути створене застосування, здатне функціонувати самостійно. З іншого боку при угрупованні процесів і даних в підсистеми необхідно враховувати вимоги до конфігурації продукту, якщо вони були сформульовані на етапі аналізу.

### **3. Програмні засоби підтримки життєвого циклу ПЗ**

#### **3.1. Методології проектування ПЗ як програмні продукти.**

Сучасні методології і технології, що реалізують їх, поставляються в електронному вигляді разом з CASE-засобами і включають бібліотеки процесів, шаблонів, методів, моделей і інших компонент, призначених для побудови ПЗ того класу систем, на який орієнтована методологія. Електронні методології включають також засоби, які повинні забезпечувати їх адаптацію для конкретних користувачів і розвиток методології за наслідками виконання конкретних проектів.

Процес адаптації полягає у видаленні непотрібних процесів, дій ЖЦ і інших компонентів методології, в зміні невідповідних або в додаванні власних процесів і дій, а також методів, моделей, стандартів і керівництва. Налаштування методології може здійснюватися також по наступних аспектах: етапи і операції ЖЦ, учасники проекту, використовувані моделі ЖЦ, підтримувані концепції і ін.

Електронні методології і технології (і CASE-засоби, що підтримують їх) складають ядро комплексу узгоджених інструментальних засобів середовища розробки ІС.

##### **3.1.1. Методологія DATARUN**

Одній з найбільш поширених в світі електронних методологій є методологія DATARUN [6,26]. Відповідно до методології DATARUN ЖЦ ПЗ розбивається на стадії, які зв'язуються з результатами виконання основних процесів, визначуваних стандартом ISO 12207. Кожну стадію окрім її результатів повинен завершувати план робіт на наступну стадію.

Стадія формування вимог і планування включає дії з визначення початкових оцінок об'єму і вартості проекту. Повинні бути сформульовані вимоги і економічне обґрунтування для розробки ІС, функціональні моделі (моделі бізнес-процесів організації) і початкова концептуальна модель даних, які дають основу для оцінки тієї, що технічної реалізовується проекту. Основними результатами цієї стадії повинні бути моделі діяльності організації (початкові моделі процесів і даних організації), вимоги до системи, включаючи вимоги по сполученню з тими, що існують ІС, початковий бізнес-план.

Стадія концептуального проектування починається з детального аналізу первинних даних і уточнення концептуальної моделі даних, після чого проектується архітектура системи. Архітектура включає розділення концептуальної моделі на осяжні підмоделі. Оцінюється можливість використання що існують ІС і вибирається відповідний метод їх перетворення. Після побудови проекту уточнюється початковий бізнес-план. Вихідними компонентами цієї стадії є концептуальна модель даних, модель архітектури системи і уточнений бізнес-план.

На стадії специфікації застосувань продовжується процес створення і деталізації проекту. Концептуальна модель даних перетвориться в реляційну модель даних. Визначається структура застосування, необхідні інтерфейси

застосування у вигляді екранів, звітів і пакетних процесів разом з логікою їх виклику. Модель даних уточнюється бизнес-правилами і методами для кожної таблиці. В кінці цієї стадії ухвалюється остаточне рішення про спосіб реалізації застосувань. За наслідками стадії повинен бути побудований проект ІС, що включає моделі архітектури ІС, даних, функцій, інтерфейсів (із зовнішніми системами і з користувачами), вимог до застосувань (моделі даних, інтерфейсів і функцій), що розробляються, вимог до доопрацювань тих, що існують ІС, вимог до інтеграції застосувань, а також сформований остаточний план створення ІС.

На стадії розробки, інтеграції і тестування повинні бути створена тестова база даних, приватні і комплексні тести. Проводиться розробка, прототипування і тестування баз даних і застосувань відповідно до проекту. Відлажуються інтерфейси з існуючими системами. Описується конфігурація поточної версії ПЗ. На основі результатів тестування проводиться оптимізація бази даних і застосувань. Застосування інтегруються в систему, проводиться тестування застосувань у складі системи і випробування системи. Основними результатами стадії є готові застосування, перевірені у складі системи на комплексних тестах, поточний опис конфігурації ПЗ, скоректована за наслідками випробувань версія системи і експлуатаційна документація на систему.

Стадія впровадження включає дії з установки і впровадження баз даних і застосувань. Основними результатами стадії повинні бути готова до експлуатації і перенесена на програмно-апаратну платформу замовника версія системи, документація супроводу і акт приймальних випробувань за наслідками дослідної експлуатації.

Стадії супроводу і розвитку включають процеси і операції, пов'язані з реєстрацією, діагностикою і локалізацією помилок, внесенням змін і тестуванням, проведенням доопрацювань, тиражуванням і розповсюдженням нових версій ПЗ в місця його експлуатації, перенесенням застосувань на нову платформу і масштабуванням системи. Стадія розвитку фактично є повторною ітерацією стадії розробки.

Методологія DATARUN спирається на дві моделі або на два уявлення:

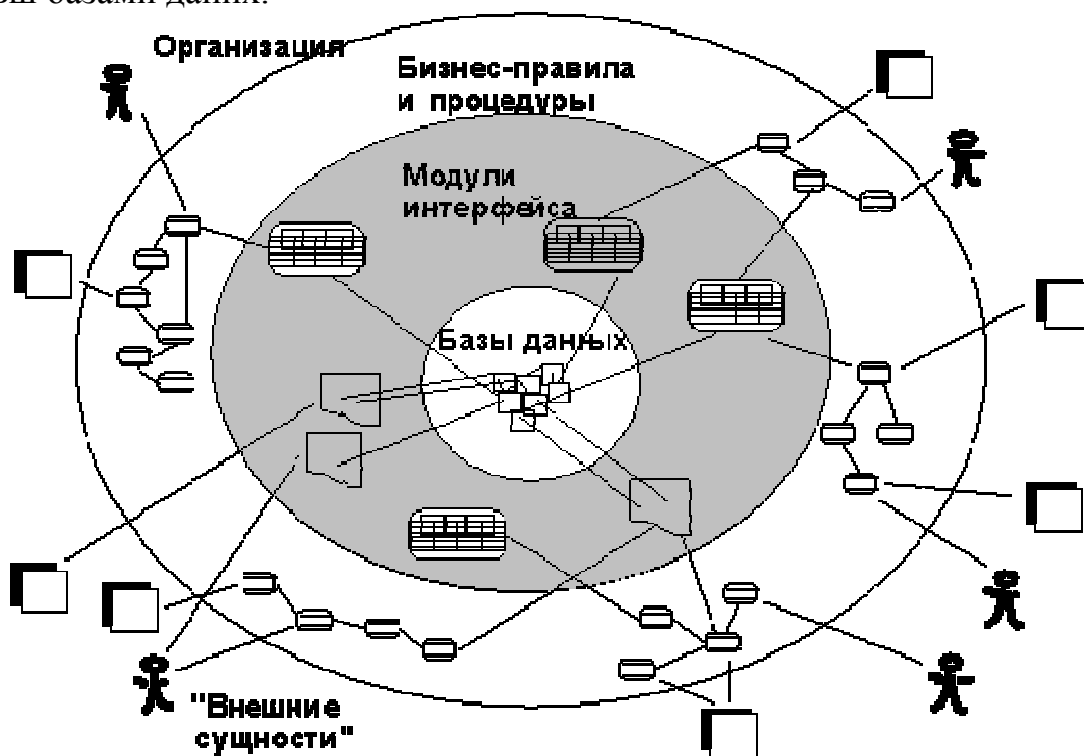
- модель організації;
- модель ІС.

Методологія DATARUN базується на системному підході до опису діяльності організації. Побудова моделей починається з опису процесів, з яких потім витягуються первинні дані (стабільна підмножина даних, які організація повинна використовувати для своєї діяльності). Первинні дані описують продукти або послуги організації, виконувані операції (транзакції) і споживані ресурси. До первинних відносяться дані, які описують зовнішню і внутрішню суть, такі як службовці, клієнти або агентства, а також дані, отримані в результаті ухвалення рішень, як наприклад, графіки робіт, ціни на продукти.

Основний принцип DATARUN полягає в тому, що первинні дані, якщо вони належним чином організовані в модель даних, стають основою для

проектування архітектури ІС. Архітектура ІС буде стабільнішою, якщо вона заснована на первинних даних, тісно пов'язаних з основними діловими операціями, що визначають природу бізнесу, а не на традиційній функціональній моделі.

Будь-яка ІС (малюнок 3.1) є набором модулів, що виконуються процесорами і що взаємодіють з базами даних. Бази даних і процесори можуть розташовуватися централізований або бути розподіленими. Події в системі можуть ініціюватися зовнішньою суттю, такими як клієнти у банкоматів або тимчасові події (кінець місяця або кварталу). Всі транзакції здійснюються через об'єкти або модулі інтерфейсу, які взаємодіють з однією або більш базами даних.



Мал. 3.1. Модель ІС

Підхід DATARUN переслідує дві мета:

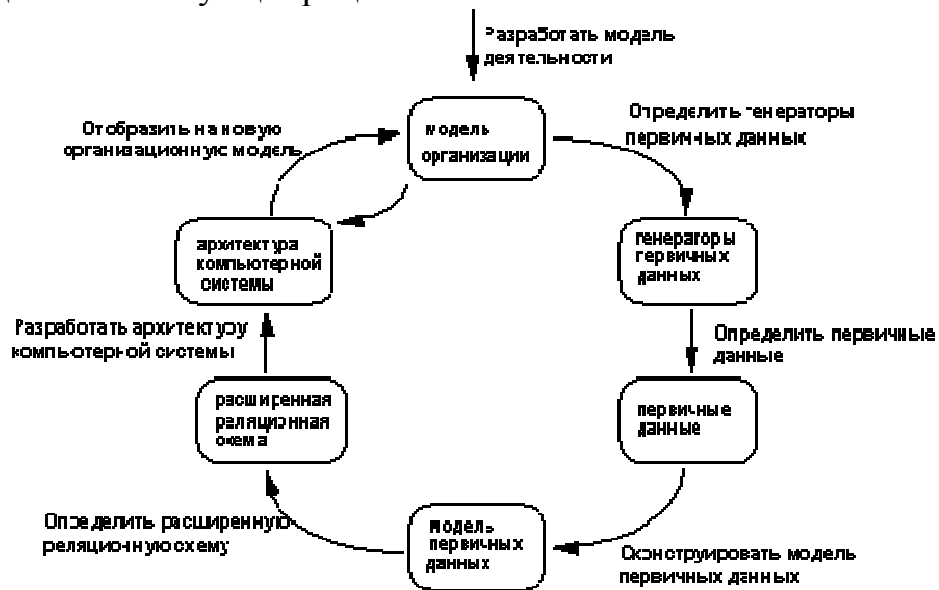
- визначити стабільну структуру, на основі якої будуватиметься ІС. Такою структурою є модель даних, отримана з первинних даних, що представляють фундаментальні процеси організації;
- спроектувати ІС на підставі моделі даних.

Об'єкти, що формуються на підставі моделі даних, є об'єктами бази даних, зазвичай розміщуваними на серверах в середовищі клієнт/сервер. Об'єкти інтерфейсу, визначені в архітектурі комп'ютерної системи, зазвичай розміщуються на клієнтській частині. Модель даних, що є основою для специфікації спільно використовуваних об'єктів бази даних і різних об'єктів інтерфейсу, забезпечує супроводжує ІС. На малюнку 3.2 представлена послідовність кроків проектування ІС.

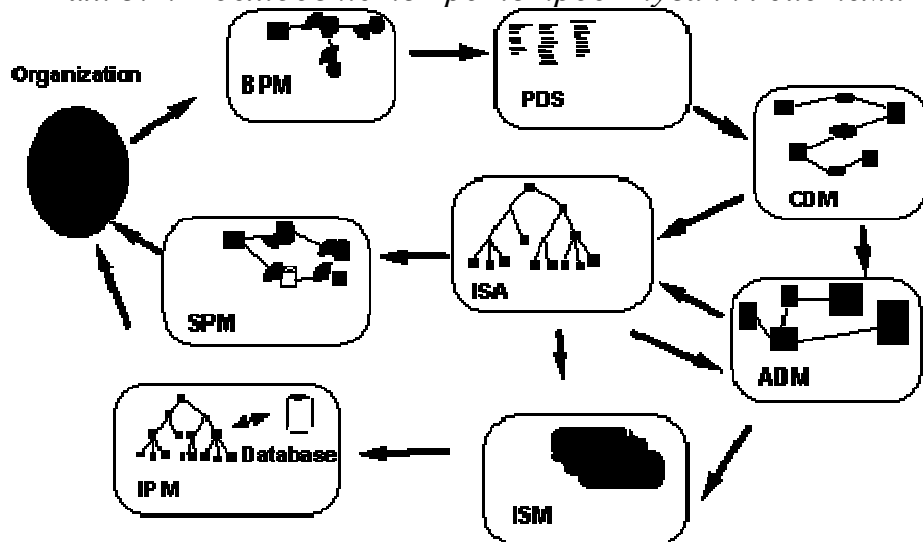
На малюнку 3.3 визначені моделі, що створюються в процесі розробки ІС. Для їх створення використовується CASE-засобів Silverrun, описаний в підрозділі 5.1. Silverrun забезпечує автоматизацію проведення проектних

робіт відповідно до методології DATARUN. Що надається цими засобами середовище проектування дає можливість керівникові проекту контролювати проведення робіт, відстежувати виконання робіт, вчасно помічати відхилення від графіка. Кожен учасник проекту, підключившись до цього середовища, може з'ясувати зміст і терміни виконання дорученої йому роботи, детально вивчити техніку її виконання в гіпертексті по технологіях, і викликати інструмент (модуль Silverrun) для реального виконання роботи.

Інформаційна система створюється послідовною побудовою ряду моделей, починаючи з моделі бізнес-процесів і закінчуючи моделлю програми, що автоматизує ці процеси.



Мал. 3.2. Послідовність кроків проектування системи



- BPM (Business Process Model)** - модель бізнес-процесів.
- PDS (Primary Data Structure)** - структура первинних даних.
- CDM (Conceptual Data Model)** - концептуальна модель даних.
- SPM (System Process Model)** - модель процесів системи.
- ISA (Information System Architecture)** - архітектура інформаційної системи.
- ADM (Application Data Model)** - модель даних застосування.
- IPM (Interface Presentation Model)** - модель представлення інтерфейсу.



**ISM (Interface Specification Model)** - модель специфікації інтерфейсу.

*Мал. 3.3. Моделі, що створюються за допомогою підходу DATARUN*

Створювана ІС повинна ґрунтуватися на функціях, що виконуються організацією. Тому перша створювана модель - це модель бізнес-процесів, побудова якої здійснюється в модулі Silverrun BPM. Для цієї моделі використовується спеціальна нотація BPM. В процесі аналізу і специфікації бізнес-функцій виявляються основні інформаційні об'єкти, які документуються як структури даних, пов'язані з потоками і сховищами моделі. Джерелами для створення структур є використовувані в організації документи, посадові інструкції, описи виробничих операцій. Ці дані вводяться в тому вигляді, як вони існують в діяльності організації. Нормалізація і видалення надмірності проводиться пізнішим при побудові концептуальної моделі даних в модулі Silverrun ERX. Після створення моделі бізнес-процесів інформація зберігається в репозиторії проекту.

В процесі обстеження роботи організації виявляються і документуються структури первинних даних. Ці структури заносяться в репозиторій модуля BPM при описі циркулюючих в організації документів, повідомлень, даних. У моделі бізнес-процесів первинні структури даних пов'язані з потоками і сховищами інформації.

На основі структур первинних даних в модулі Silverrun ERX створюється концептуальна модель даних (ER-модель). Від структур первинних даних концептуальна модель відрізняється видаленням надмірності, стандартизацією найменувань понять і нормалізацією. Ці операції в модулі ERX виконуються за допомогою вбудованої експертної системи. Мета концептуальної моделі даних - описати використовувану інформацію без деталей можливої реалізації в базі даних, але в добре структурованому нормалізованому вигляді.

На основі моделі бізнес-процесів і концептуальної моделі даних проектується архітектура ІС. Визначаються вхідні в систему застосування, для кожного застосування специфікуються використовувані дані і функції, що реалізуються. Архітектура ІС створюється в модулі Silverrun BPM з використанням спеціальної нотації ISA. Основний зміст цієї моделі - структурні компоненти системи і навігація між ними. Концептуальна модель даних розбивається на частини, відповідні вхідним до складу системи застосуванням.

Перед розробкою застосувань повинна бути спроектована структура корпоративної бази даних. DATARUN припускає використання бази даних, заснованої на реляційній моделі. Концептуальна модель даних після нормалізації переноситься в модуль реляційного моделювання Silverrun RDM за допомогою спеціального моста ERX-RDM. Перетворення моделі з формату ERX у формат RDM відбувається автоматично без втручання користувача. Після перетворення форматів виходить модель реляційної бази даних. Ця модель деталізує в модулі Silverrun RDM визначенням фізичної реалізації (типів даних СУБД, ключів, індексів, тригерів, обмежень посилальної цілісності). Правила обробки даних можна задавати як

безпосередньо на мові програмування СУБД, так і в декларативній формі, не прив'язаній до реалізації. Мости Silvergun до реляційних СУБД перекладають ці декларативні правила мовою необхідної системи, що знижує трудомісткість програмування процедур сервера бази даних, а також дозволяє з однієї специфікації генерувати застосування для різних СУБД.

За допомогою моделі системних процесів детально документується поведінка кожного застосування. У модулі BPM створюється модель системних процесів, що визначає, яким чином реалізуються бизнес-процеси. Ця модель створюється окремо для кожного застосування і тісно пов'язана з моделлю даних застосування.

Застосування складається з інтерфейсних об'єктів (екранних форм, звітів, процедур обробки даних). Кожен інтерфейс системи (екранна форма, звіт, процедура обробки даних) має справу з підмножиною бази даних. У моделі даних застосування (створеною в модулі RDM) створюється підсхема бази даних для кожного інтерфейсу цього застосування. Уточнюються також правила обробки даних, специфічні для кожного інтерфейсу. Інтерфейс працює з даними в ненормалізованому вигляді, тому специфікація даних, як її бачить інтерфейс, оформляється як окрема підсхема моделі даних інтерфейсу.

Модель представлення інтерфейсу - це опис зовнішнього вигляду інтерфейсу, як його бачить кінцевий користувач системи. Це може бути як документ, що показує зовнішній вигляд екрану або структуру звіту, так і сам екран (звіт), створений за допомогою одного із засобів візуальної розробки застосувань - так званих мов четвертого покоління (4GL - Fourth Generation Languages). Оскільки більшість мов 4GL дозволяють швидко створювати працюючі прототипи застосувань, користувач має можливість побачити працюючий прототип системи на ранніх стадіях проектування.

Після створення підсхем реляційної моделі для застосувань проектується детальна структура кожного застосування у вигляді схеми навігації екранів, звітів, процедур пакетної обробки. На даному кроці ця структура деталізує до вказівки конкретних стовпців і таблиць бази даних, правил їх обробки, виду екранних форм і звітів. Отримана модель детально документує застосування і безпосередньо використовується для програмування специфікованих інтерфейсів.

Далі, за допомогою засобів розробки застосувань відбувається фізичне створення системи: застосування програмуються і інтегруються в інформаційну систему.

### **3.1.2. Інструментальний засіб SE Companion**

Інструментальний засіб SE Companion [27] є середовищем, в якому реалізований електронний варіант методології DATARUN. Воно дозволяє:

- створити гіпертекстовий опис методології у вигляді ієрархії опису стадій, етапів і операцій розробки;
- створити гіпертекстовий опис всіх методів і методик реалізації процесів ЖЦ ПЗ;
- виділити з гіпертекстового опису ієрархію процесів ЖЦ ПЗ для

планування і управління процесом створення ПЗ (ієрархію робіт);

- змінювати гіпертекстові описи ЖЦ і методів так, як це необхідно розробникові, іншими словами, проводити авторизацію методології і відстежувати ці зміни в ієрархії робіт, призначеній для управління проектом;

- прив'язати до процесів ЖЦ інструментальні засоби підтримки цих процесів і забезпечити виклик інструментальних засобів з відповідних екранів гіпертекстового довідника;

- забезпечити проглядання гіпертекстових екранів опису використовуваних методів з інструментальних засобів;

- забезпечити підтримку процесу управління розробкою, зокрема, за рахунок взаємодії із засобом планування робіт MS Project, оцінювання трудомісткості проекту, відстежування виконання робіт, створення графіків робіт, і ін.

Особливо важливими є можливість авторизації методології і інтерактивний доступ будь-якого розробника до опису будь-якого методу або процесу в потрібний йому момент часу. На сучасному етапі розвитку технології, в умовах швидкої зміни як програмних і апаратних засобів, так і завдань бізнесу, методологія створення, супроводу і розвитку ПЗ не повинна бути незмінною; вона повинна мати можливість змінюватися і настроюватися на нові технології, методи і інструментальні засоби. Сучасні розробники великих ІС набувають однієї або декілька методологій постачальника, а потім створюють на їх основі власні методології і технології, адаптовані до конкретних умов (див. підрозділ [1.3](#)).

У SE Companion початковим документом, що описує методологію (як процеси ЖЦ, так і всі супутні методи і методики), є файл у форматі MS Word. Це забезпечує можливості для опису методології з будь-яким ступенем деталізації, проведення розмітки для створення гіпертексту і авторизації методології в прийнятому стандартному форматі.

Гіпертекстовий опис методології і технології створення ПЗ будується з опису процесів життєвого циклу, методів і методик, і є єдиним гіпертекстовим документом у форматі MS Help. Підсумковий гіпертекстовий опис виходить в результаті трансляції початкового документа. Всі зміни і доповнення методології проводяться за допомогою коректування і, можливо, додаткової розмітки початкового документа.

Опис методології створення системи зазвичай складається з розділу опису процесів ЖЦ і розділів опису методів і методик. У свою чергу, розділ описів процесів складається з ієрархії описів стадій, етапів і операцій життєвого циклу з обов'язковим описом вихідних компонентів кожного процесу. Компоненти ПЗ створюються із застосуванням методик і методів, що описуються у відповідних розділах.

Мінімальна конфігурація апаратно-програмних засобів, необхідних SE Companion Authoring Tool:

- процесор: i486/33;

- оперативна пам'ять: 4 Мбайт для проглядання гіпертексту, 12 Мбайт

для авторизації;

- дискова пам'ять: 20 Мбайт;
- операційні середовища: Microsoft Windows 3.1, Microsoft Windows for Workgroups 3.11, Microsoft Windows NT 3.5, Microsoft Windows 95.

### **3.2. CASE-засоби. Загальна характеристика і класифікація**

Сучасні CASE-засоби охоплюють обширну область підтримки численних технологій проектування ІС: від простих засобів аналізу і документування до повномасштабних засобів автоматизації, що покривають весь життєвий цикл ПЗ.

Найбільш трудомісткими етапами розробки ІС є етапи аналізу і проектування, в процесі яких CASE-засоби забезпечують якість ухвалюваних технічних рішень і підготовку проектної документації. При цьому велику роль грають методи візуального представлення інформації. Це припускає побудову структурних або інших діаграм в реальному масштабі часу, використання багатообразної колірної палітри, крізь перевірку синтаксичних правил. Графічні засоби моделювання наочної області дозволяють розробникам в наочному вигляді вивчати що існує ІС, перебудовувати її відповідно до поставлених цілей і наявних обмежень.

У розряд CASE-засобів потрапляють як відносно дешеві системи для персональних комп'ютерів з вельми обмеженими можливостями, так і дорогі системи для неоднорідних обчислювальних платформ і операційних середовищ. Так, сучасний ринок програмних засобів налічує близько 300 різних CASE-засобів, найбільш могутні з яких так чи інакше використовуються практично всіма ведучими західними фірмами.

Зазвичай до CASE-засобів відносять будь-який програмний засіб, що автоматизує ту або іншу сукупність процесів життєвого циклу ПЗ і що володіє наступними основними характерними особливостями:

- могутні графічні засоби для опису і документування ІС, що забезпечують зручний інтерфейс з розробником і розвивають його творчі можливості;
- інтеграція окремих компонент CASE-засобів, що забезпечує керованість процесом розробки ІС;
- використання спеціальним чином організованого сховища проектних метаданих (репозиторія).

Інтегрований CASE-засіб (або комплекс засобів, що підтримують повний ЖЦ ПЗ) містить наступні компоненти;

- репозиторій, що є основою CASE-засобу. Він повинен забезпечувати зберігання версій проекту і його окремих компонентів, синхронізацію надходження інформації від різних розробників при груповій розробці, контроль метаданих на повноту і несуперечність;
- графічні засоби аналізу і проектування, забезпечуюче створення і редагування ієрархічно зв'язаних діаграм (DFD, ERD і ін.), створюючих моделі ІС;
- засоби розробки застосувань, включаючи мови 4GL і генератори

код;

- засоби конфігураційного управління;
- засоби документування;
- засоби тестування;
- засоби управління проектом;
- засоби реінжинірингу.

Вимоги до функцій окремих компонент у вигляді критеріїв оцінки CASE-засобів наведені в розділі [4.2.](#)

Всі сучасні CASE-засоби можуть бути класифіковані в основному по типах і категоріях. Класифікація по типах відображає функціональну орієнтацію CASE-засобів на ті або інші процеси ЖЦ. Класифікація по категоріях визначає ступінь інтегрованості по виконуваних функціях і включає окремі локальні засоби, вирішальні невеликі автономні завдання (tools), набір частково інтегрованих засобів, що охоплюють більшість етапів життєвого циклу ІС (toolkit) і повністю інтегровані засоби, що підтримують весь ЖЦ ІС і зв'язані загальним репозиторієм. Крім цього, CASE-засоби можна класифікувати по наступних ознаках:

- вживаним методологіям і моделям систем і БД;
- ступені інтегрованості з СУБД;
- доступним платформам.

Класифікація по типах в основному співпадає з компонентним складом CASE-засобів і включає наступні основні типи:

- засоби аналізу (Upper CASE), призначені для побудови і аналізу моделей наочної області (Design/IDEF (Meta Software), BPwin (Logic Works));

- засоби аналізу і проектування (Middle CASE), підтримуючі найбільш поширені методології проектування і що використовуються для створення проектних специфікацій (Vantage Team Builder (Cayenne), Designer/2000 (ORACLE), Silverrun (CSA), PRO-IV (McDonnell Douglas), CASE.Аналитик (Макропроджект)). Виходом таких засобів є специфікації компонентів і інтерфейсів системи, архітектура системи, алгоритмів і структур даних;

- засоби проектування баз даних, що забезпечують моделювання даних і генерацію схем баз даних (як правило, на мові SQL) для найбільш поширених СУБД. До них відносяться ERwin (Logic Works), S-Designor (SDP) і DataBase Designer (ORACLE). Засоби проектування баз даних є також у складі CASE-засобів Vantage Team Builder, Designer/2000, Silverrun і PRO-IV;

- засоби розробки застосувань. До них відносяться засоби 4GL (Uniface (Compuware), JAM (JYACC), PowerBuilder (Sybase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Gupta), Delphi (Borland) і ін.) і генератори код, що входять до складу Vantage Team Builder, PRO-IV і частково - в Silverrun;

- засоби реінжиніринга, що забезпечують аналіз програмних код і схем баз даних і формування на їх основі різних моделей і проектних

специфікацій. Засоби аналізу схем БД і формування ERD входять до складу Vantage Team Builder, PRO-IV, Silverrun, Designer/2000, ERwin і S-Designor. В області аналізу програмних код найбільшого поширення набувають об'єктно-орієнтовані CASE-засоби, що забезпечують реінжиниринг програм на мові С++ (Rational Rose (Rational Software), Object Team (Cayenne)).

Допоміжні типи включають:

- засоби планування і управління проектом (SE Companion, Microsoft Project і ін.);
- засоби конфігураційного управління (PVCS (Intersolv));
- засоби тестування (Quality Works (Segue Software));
- засоби документування (SODA (Rational Software)).

На сьогоднішній день Російський ринок програмного забезпечення має в своєму розпорядженні наступні найбільш розвинені CASE-засобами:

- Vantage Team Builder (Westmount I-CASE);
- Designer/2000;
- Silverrun;
- ERwin+BPwin;
- S-Designor;
- CASE.Аналитик.

Опис перерахованих CASE-засобів приведений в розділі 5. Крім того, на ринку постійно з'являються як нові для вітчизняних користувачів системи (наприклад, CASE /4/0, PRO-IV, System Architect, Visible Analyst Workbench, EASYCASE), так і нові версії і модифікації перерахованих систем.

#### 4. Технологія впровадження CASE-засобів

Приведена в даному розділі технологія базується в основному на стандартах IEEE [16,17] (IEEE - Institute of Electrical and Electronics Engineers - Інститут інженерів по електротехніці і електроніці). Термін "впровадження" використовується в широкому сенсі і включає всі дії від оцінки первинних потреб до повномасштабного використання CASE-засобів в різних підрозділах організації-користувача. Процес впровадження CASE-засобів складається з наступних етапів [16]:

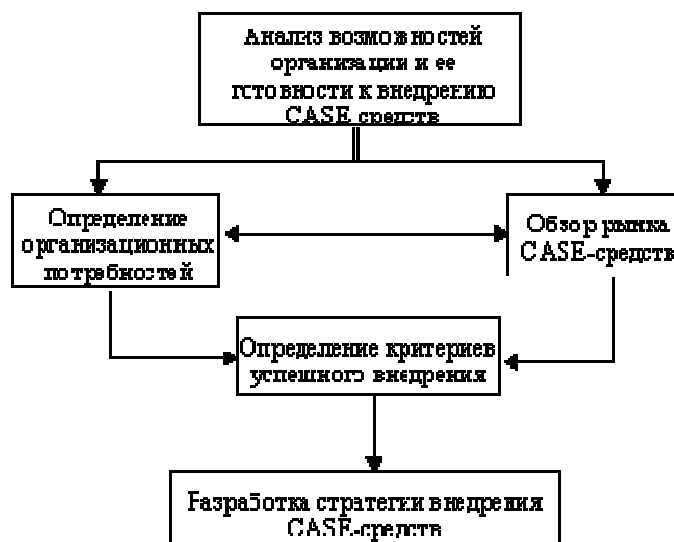
- визначення потреб в CASE-засобах;
- оцінка і вибір CASE-засобів;
- виконання пілотного проекту;
- практичне впровадження CASE-засобів.

Процес успішного впровадження CASE-засобів не обмежується тільки їх використанням. Насправді він охоплює планування і реалізацію безлічі технічних, організаційних, структурних процесів, змін в загальній культурі організації, і заснований на чіткому розумінні можливостей CASE-засобів.

На спосіб впровадження CASE-засобів може вплинути специфіка конкретної ситуації. Наприклад, якщо замовник віддає перевагу конкретному засобу, або воно обмовляється вимогами контракту, етапи впровадження повинні відповідати такому зумовленому вибору. У інших ситуаціях відносна простота або складність засобу, ступінь узгодженості або конфліктності з процесами, що існують в організації, необхідний ступінь інтеграції з іншими засобами, досвід і кваліфікація користувачів можуть привести до внесення відповідних коректив до процесу впровадження.

##### 4.1. Визначення потреб в CASE-засобах

Даний етап (малюнок 4.1) включає досягнення розуміння потреб організації і технології подальшого процесу впровадження CASE-засобів. Він повинен привести до виділення тих областей діяльності організації, в яких застосування CASE-засобів може принести реальну користь. Результатом даного етапу є документ, що визначає стратегію впровадження CASE-засобів.



## *Мал. 4.1. Визначення потреб в CASE-засобах*

### **4.1.1. Аналіз можливостей організації**

Першою дією даного етапу є аналіз можливостей організації відносно її технологічної бази, персоналу і використовуваного ПЗ. Такий аналіз може бути формальним або неформальним.

Формальні підходи визначаються моделлю оцінки зрілості технологічних процесів організації CMM (Capability Maturity Model), розробленою SEI (Software Engineering Institute), а також стандартами ISO 9001: 1994, ISO 9003-3: 1991 і ISO 9004-2:1991. В центрі уваги цих підходів знаходиться аналіз різних аспектів процесів, що відбуваються в організації.

Для отримання інформації щодо положення і потреб організації можуть використовуватися неформальні оцінки і анкетування. Список простих питань, які можуть допомогти в неформальній оцінці поточної практики використання ПЗ, технології і персоналу приведений нижче.

Відповіді на дані питання можуть визначити ті області, де автоматизація може принести ефект. Інакше може опинитися, що вдосконалення процесу розробки і супроводу ПЗ, програм навчання і інших функцій переважніше, ніж придбання нових засобів. Деякі з цих удосконалень можуть виявитися необхідними для отримання максимальної вигоди від впровадження будь-яких засобів.

Дані питання є, по суті, керівництвом по збору інформації, необхідної для визначення ступеня готовності організації до впровадження CASE-технології.

#### *Загальні питання*

- використовується модель ЖЦ (каскадна або спіральна);
- використовувані методи (структурні, об'єктно-орієнтовані). Ступінь адаптації методу до потреб організації; кваліфікація співробітників;
- наявність документованих стандартів (формальних або неформальних) по аналізу вимог, специфікаціям і проектуванню, кодуванню і тестуванню;
- кількісні метрики, використовувані в процесі розробки ПЗ, їх використання;
- види документації, що випускається в процесі ЖЦ ПЗ;
- наявність групи підтримки засобів проектування.

#### *Проекти, що ведуться в організації*

- середня тривалість проекту в людино-місяцях;
- середня кількість фахівців, що беруть участь в проектах різних категорій (невеликих, середніх і великих);
- середній розмір проектів різних категорій в термінах кодових метрик (наприклад, в рядках початкових код), спосіб вимірювання.

#### *Технологічна база*

Технологічна база організації включає не тільки технічні засоби, використовувані при розробці ПЗ, але також мови, засоби, методи і середовище функціонування ПЗ. Ця база дуже істотно впливає на вибір



відповідних CASE-засобів. Питання, що стосуються технології, включають наступні:

- доступні обчислювальні ресурси, платформа розробки;
- рівень доступності ресурсів, вузькі місця, середній час очікування ресурсів;
- ПО, використовуване в організації, і його характер (готові програмні продукти, власні розробки);
- ступінь інтеграції використовуваних програмних продуктів, механізми інтеграції (що існують і плановані);
- тип і рівень мережеских можливостей, доступних групі розробників;
- використовувані мови програмування;
- середній відсоток тих, що знов розробляються, повторно використовуваних і реально експлуатованих застосувань.

#### *Персонал*

Головною метою оцінки персоналу є визначення його відношення до можливих змін (позитивного, нейтрального або негативного). Питання, що стосуються оцінки персоналу, включають наступні:

- реакція співробітників організації (як окремих людей, так і колективів) на впровадження нової технології. Наявність досвіду успішних або безуспішних впроваджень;
- наявність лідерів, здатних серйозно вплинути на відношення до нових засобів;
- наявність прагнення "знизу" до вдосконалення засобів і технології;
- об'єм навчання, необхідного для орієнтації користувачів в новій технології;
- стабільність і рівень текучості кадрів.

#### *Готовність*

Метою оцінки готовності організації є визначення того, наскільки вона здатна сприйняти як негайні, так і довгострокові наслідки впровадження CASE-засобів. Питання, що стосуються оцінки готовності, включають наступні:

- підтримка проекту з боку вищого керівництва;
- готовність організації до довгострокового фінансування проекту;
- готовність організації до виділення необхідних фахівців для участі в процесі впровадження і до їх навчання;
- готовність персоналу до істотної зміни технології своєї роботи;
- ступінь розуміння персоналом масштабу змін;
- готовність технічних фахівців і менеджерів піти на можливе короткочасне зниження продуктивності своєї роботи;
- готовність керівництва до довготривалого очікування віддачі від вкладених засобів.

Оцінка готовності організації до впровадження CASE-технології повинна бути відвертою і ретельною, оскільки у разі відсутності такої готовності всі зусилля по впровадженню потерплять крах.

#### **4.1.2. Визначення організаційних потреб**

Організаційні потреби виходять безпосередньо з проблем організації і цілей, які вона прагне досягти. Проблеми і цілі можуть бути пов'язані з управлінням, виробництвом продукції, економікою, персоналом або технологією. Питання, що стосуються визначення цілей, потреб і очікуваних результатів, приведені нижче. Визначення потреб повинне виконуватися у поєднанні з оглядом ринку CASE-засобів, оскільки інформація про технології, доступні на ринку в даний момент, може зробити вплив на потреби.

##### *Цілі організації*

Цілі організації грають головну роль у визначенні її конкретних потреб і очікуваних результатів. Для їх розуміння необхідно відповісти на наступні питання:

- намір організації використовувати CASE-технологію для допомоги в досягненні певної мети або очікувань (наприклад, певного рівня CMM або сертифікації відповідно до ISO 9001);
- сприйняття CASE-технології як чинника, сприяючого досягненню стратегічних цілей організації;
- наявність у організації власної програми вдосконалення процесу розробки ПЗ;
- сприйняття ініціативи впровадження CASE-технології як частини більш широкомасштабного проекту по створенню середовища розробки ПЗ.

##### *Потреби організації*

Визначення потреб організації, зв'язаних з використанням CASE-технології, включає аналіз цілей і існуючих можливостей. Після того, як всі потреби організації визначені, кожною з них повинен бути привласнений певний пріоритет, що відображає її значущість для успішної діяльності організації. Якщо потреби, пов'язані з CASE-технологією, не володіють вищим пріоритетом, має сенс відмовитися від її впровадження і зосередитися на потребах з найвищим пріоритетом.

Доцільно побудувати матрицю відповідності потреб організації можливостям основних CASE-засобів. Складання такої матриці вимагає певного рівня знань ринку CASE-засобів. Кінець кінцем кожна функція або можливість засобу повинні точно відповідати деякій потребі з певним пріоритетом.

Визначенню потреб організації можуть допомогти відповіді на наступні питання:

- яким чином продуктивність і якість діяльності організації порівнюються з аналогічними показниками подібних організацій (на жаль, багато організацій не мають в своєму розпорядженні даних для такого порівняння);
- які процеси ЖЦ ПЗ дають якнайкращу (і, відповідно, якнайгіршу) віддачу; чи існують конкретні процеси, які можуть бути вдосконалені шляхом використання нових методів і засобів.

### *Очікувані результати*

З впровадженням CASE-засобів зазвичай зв'язують великі очікування. У ряді випадків ці очікування виявляються нереалістичними і приводять до невдачі при впровадженні.

Складання реалістичного переліку очікуваних результатів є важким завданням, оскільки він може залежати від таких чинників, як тип упроваджуваних засобів і характеристики впроваджувальної організації.

Ряд потенційно реалістичних і нереалістичних очікуваних результатів, пов'язаних з організацією в цілому, користувачами, плануванням, аналізом, проектуванням, розробкою і витратами, приведені нижче. Практично неможливо, щоб в процесі одного впровадження CASE-засобів були досягнуті всі позитивні результати. Проте, будь-яка організація може виробити власний підхід до очікуваних результатів, маючи на увазі, що даний перелік є всього лише прикладом.

#### Реалістичні очікування:

- підвищення уваги до планування діяльності, пов'язаної з інформаційною технологією;
- підтримка реінжиніринга бизнес-процесов;
- довготривале підвищення продуктивності і якості діяльності організації;
- прискорення і підвищення узгодженості розробки застосувань;
- зниження частки ручної праці в процесі розробки і/або експлуатації;
- точніша відповідність застосувань вимогам користувачів;
- відсутність необхідності великої переробки застосувань для підвищення їх ефективності;
- поліпшення реакції служби експлуатації на вимоги внесення змін і удосконалень;
- підвищення якості документування;
- поліпшення комунікації між користувачами і розробниками;
- послідовне і постійне підвищення якості проектування;
- вищі можливості повторного використання розробок;
- короткочасне зростання витрат, пов'язане з діяльністю по впровадженню CASE-засобів;
- послідовне зниження загальних витрат;
- поліпшення прогнозованості витрат.

#### Нереалістичні очікування:

- відсутність дії на загальну культуру і розподіл ролей в організації;
- розуміння проектних специфікацій непідготовленими користувачами;
- скорочення персоналу, пов'язаного з інформаційною технологією;
- зменшення ступеня участі в проектах вищого керівництва і менеджерів, а також експертів наочної області, зменшення ступеня участі користувачів в процесі розробки застосувань;
- негайне підвищення продуктивності діяльності організації;

- досягнення абсолютної повноти і несуперечності специфікацій;
- автоматична генерація прикладних систем з проектних специфікацій;
- негайне зниження витрат, пов'язаних з інформаційною технологією;
- зниження витрат на навчання.

Реалізм в оцінці очікуваних витрат має особливо важливе значення, оскільки він дозволяє правильно оцінити віддачу від інвестицій. Витрати на впровадження CASE-засобів зазвичай недооцінюються. Серед конкретних статей витрат на впровадження можна виділити наступні:

- фахівці з планування впровадження CASE-засобів;
- вибір і установка;
- облік специфічних вимог персоналу;
- придбання CASE-засобів і навчання;
- настройка;
- підготовка документації, стандартів і процедур використання засобів;
- інтеграція з іншими засобами і існуючими даними;
- освоєння засобів розробниками;
- технічні засоби;
- оновлення версій.

Важливо також усвідомлювати, що поліпшення діяльності організації, що є наслідком використання CASE-технології, може бути неочевидним протягом найпершого проекту, що використовує нову технологію. Продуктивність і інші характеристики діяльності організації можуть спочатку навіть погіршити, оскільки на освоєння нових засобів і внесення необхідних змін в процес розробки потрібний якийсь час. Таким чином, очікувані результати повинні розглядатися з урахуванням вірогідного відстрочення в поліпшенні проектних характеристик.

Кожна потреба повинна мати певний пріоритет, залежний від того, наскільки критичною вона є для досягнення успіху в організації. Кінець кінцем, повинна чітко простежуватися дія кожної функції або можливості засобів, що набувають, на задоволення конкретних потреб.

Результатом даної дії є формулювання потреб з їх пріоритетами, яка використовується на етапі оцінки і вибору як "призначені для користувача потреби".

#### **4.1.3. Аналіз ринку CASE-засобів**

Потреби організації в CASE-засобах повинні розміряться з реальною ситуацією на ринку або власними можливостями розробки. Дослідження ринку проводиться шляхом вивчення літератури по CASE-засобів, відвідини конференцій і семінарів, що проводяться постачальниками (їх перелік приведений в кінці даного огляду) і користувачами CASE-засобів. При проведенні даного аналізу необхідно з'ясувати можливість інтеграції конкретного CASE-засобу з іншими засобами, використовуваними (або планованими до використання) організацією. Крім того, важливо отримати

достовірну інформацію про засоби, засновану на реальному призначеному для користувача досвіді і відомостях від призначених для користувача груп.

#### **4.1.4. Визначення критеріїв успішного впровадження**

Визначувані критерії повинні дозволяти кількісно оцінювати ступінь задоволення кожній з потреб, пов'язаних з впровадженням. Крім того, по кожному критерію повинне бути визначене його конкретне оптимальне значення. На певних етапах впровадження ці критерії повинні аналізуватися для того, щоб визначити поточний ступінь задоволення потреб.

Як правило, більшість організацій здійснює впровадження CASE-засобів для того, щоб підвищити продуктивність процесів розробки і супроводу ПЗ, а також якість результатів розробки. Проте, ряд організацій не займаються і не займалися раніше збором кількісних даних по вказаних параметрах. Відсутність таких даних утрудняє кількісну оцінку дії, що надається впровадженням CASE-засобів. В цьому випадку рекомендується розробка відповідних метрик. Інформація про такі метрики приведена в стандартах IEEE Std 1045-1992 (IEEE Standard for Software Productivity Metrics) і IEEE Std 1061-1992 (IEEE Standard for a Software Quality Metrics Methodology).

В тому випадку, якщо базові метричні дані відсутні, організація часто може витягувати корисну інформацію зі своїх проектних архівів.

Крім продуктивності і якості, корисну інформацію про стан впровадження CASE-засобів також можуть дати і інші характеристики організаційних процесів і персоналу. Наприклад, оцінка ступеня успішності впровадження може включати відсоток проектів, що використовують CASE-засоби, рейтингові оцінки рівня кваліфікації фахівців, зв'язані з використанням CASE-засобів і результати опитів персоналу з приводу відношення до використання CASE-засобів. Інші приклади проектних характеристик, які можуть бути оцінені кількісно, включають наступні:

- узгодженість проектних результатів;
- точність вартісних і планових оцінок;
- мінливість зовнішніх вимог;
- дотримання стандартів організації;
- ступінь повторного використання існуючих компонентів ПЗ;
- об'єм і види необхідного навчання;
- типи і моменти виявлення проектних помилок;
- обчислювальні ресурси, використовувані CASE-засобами.

#### **4.1.5. Розробка стратегії впровадження CASE-засобів**

Стратегія впровадження повинна забезпечувати задоволення потреб і критеріїв, визначених раніше. Стратегія включає наступні складові:

- організаційні потреби;
- базові метрики, необхідні для подальшого порівняння результатів;
- критерії успішного впровадження, пов'язані із задоволенням організаційних потреб, включаючи очікувані результати послідовних етапів процесу впровадження;

- підрозділи організації, в яких повинне виконуватися впровадження CASE-засобів;
- вплив, що надається на інші підрозділи організації;
- стратегії і плани оцінки і вибору, пілотного проектування і переходу до повномасштабного впровадження;
- основні чинники ризику;
- орієнтовний рівень витрат і джерела фінансування процесу впровадження CASE-засобів;
- ключовий персонал і інші ресурси.

Необхідно відзначити, що впровадження нової технології може включати важливі і складні зміни в культурі організації. Істотна увага повинна приділятися ролям різних груп, залучених в процес таких змін. Найбільш істотні ролі включають наступні:

- спонсор (зазвичай з числа менеджерів вищого рівня). Дана роль є критичною для підтримки проекту і забезпечення необхідного фінансування. Спонсор повинен володіти чітким розумінням необхідності серйозних зусиль, пов'язаних з впровадженням CASE-засобів, і тривалості періоду очікування відчутних результатів;

- виконавець - звичайна особа (або група осіб), що усвідомлює потенційні можливості нової технології, має авторитет серед технічного персоналу і здатне очолити процес впровадження нової технології;

- цільова група - зазвичай включає менеджерів і технічний персонал, які будуть повернуті до безпосереднього використання CASE-засобів, а також фахівців, які будуть повернуті побічно, таких, як фахівці з документування, персонал підтримки мережі і замовники. Повинні бути визначені потреби кожній з таких груп і план їх ефективного задоволення.

У загальному випадку, впровадження CASE-засобів повинне управлятися і фінансуватися таким же чином, як і будь-який проект розробки ПЗ. Стратегія впровадження може бути переглянута у разі появи додаткової інформації.

Існує декілька підходів до розробки стратегії впровадження CASE-засобів. Відносні переваги того або іншого підходу перед іншими повинні розглядатися в контексті специфіки конкретної організації. Особливе значення при цьому надається персоналу організації і процесу розробки ПЗ.

Низхідний підхід до розробки стратегії визнає важливість дослідження всіх типів CASE-засобів і документування процесів розробки і супроводу ПЗ в даній організації до того, як визначаються вимоги до CASE-засобів. При цьому виконується загальний аналіз процесу створення і супроводу ПЗ в організації. Даний підхід часто спричиняє за собою загальну реорганізацію процесів створення і супроводу ПЗ в тому ступені, в якій це пов'язано з CASE-засобами. Результатом такої реорганізації стає великомасштабна стратегія автоматизації процесів створення і супроводу ПЗ.

Перевага низхідного підходу полягає в тому, що він охоплює всі процеси створення і супроводу ПЗ, забезпечуючи максимально можливу їх

автоматизацію. Іншою перевагою є придбання інтегрованого (або інтегрованого) набору засобів, оскільки кожне окреме постачання підкоряється загальній стратегії. Низхідний підхід також може бути легко інтегрований в загальну стратегію розвитку процесу створення і супроводу ПЗ, в якій впровадження CASE-засобів є тільки одним з аспектів.

Недоліки даного підходу полягають в наступному:

- низхідний підхід вимагає для своєї реалізації значних людських і фінансових ресурсів;

- у загальному випадку, широкомасштабний підхід такого роду не дозволяє користувачам достатньо швидко приступити до практичного використання засобів;

- низхідний підхід може привести до відносно серйозним змінам процесів, що існують в організації. Реалізацією такого підходу важче управляти, і, крім того, він містить в собі підвищений ризик провалу, ведучого до того, що CASE-засобів "кладуться на полку".

Низхідний підхід рекомендується для щодо зрілих організацій із сталим процесом створення і супроводу ПЗ, які прагнуть вкласти всі необхідні ресурси в повністю закінчену роботу. Щоб підвищити вірогідність успіху, потрібне ухвалення серйозних зобов'язань з боку як керівництва, так і потенційних користувачів.

Висхідний підхід починається з визначення деякого засобу або типу засобів, які потенційно можуть допомогти організації в поліпшенні виконання поточної роботи. Організація може потім оцінити можливу дію засобів на процес розробки і супроводу ПЗ.

Переваги даного підходу полягають в наступному:

- невелика автоматизація може бути виконана при мінімальних витратах;

- автоматизація може бути виконана за короткий проміжок часу, дозволяючи швидко усунути відомі недоліки в існуючих процесах;

- невеликий масштаб висхідної стратегії дозволяє краще фокусувати і контролювати дію, що надається на існуючі процеси.

- Недоліки даного підходу полягають в наступному:

- засоби, що набувають як результат окремих узятих застосувань даного підходу, можуть погано інтегруватися між собою. Це може привести до необхідності виконання великого об'єму ручної роботи;

- тоді як конкретні, порівняно невеликі проблеми вирішуються достатньо швидко, до вирішення фундаментальних проблем, пов'язаних з широким докола процесів розробки ПЗ, справа зазвичай не доходить.

Висхідний підхід рекомендується для організацій з вузько специфічними потребами в автоматизації, процесів, що не потребують загального вдосконалення. В деяких випадках може виявитися не дуже практичним приступати до такого вдосконалення, не визначивши найнасушніші потреби в автоматизації. Тоді як даний підхід може допомогти організації задовольнити найнасушніші потреби і розвинути основні процеси,

залишається істотна небезпека того, що вибраний засіб не надасть істотної дії на такі чинники, як якість і продуктивність.

Найбільш раціональна стратегія може поєднувати характеристики обох підходів. Наприклад, низхідні методи можуть використовуватися для визначення стандартів якості організації, потреб в засобах і очікуваних результатів, тоді як висхідні методи можуть використовуватися для оцінки і вибору конкретних CASE-засобів, розробки планів впровадження і контролю його результатів.

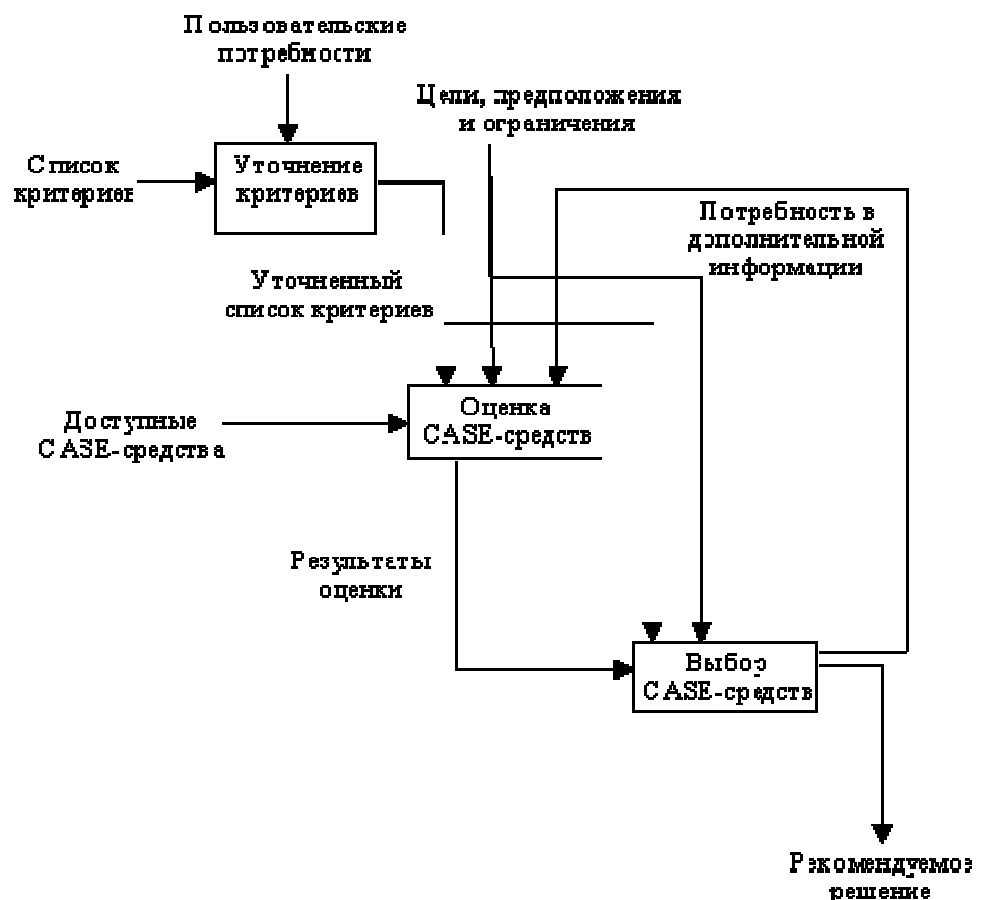
## 4.2. Оцінка і вибір CASE-засобів

### 4.2.1. Загальні відомості

Модель процесу оцінки і вибору [17], що розглядається нижче (малюнок 4.2), описує найбільш загальну ситуацію оцінки і вибору, а також показує залежність між ними. Як можна бачити, оцінка і вибір можуть виконуватися незалежно один від одного або разом, кожен з цих процесів вимагає застосування певних критеріїв.

Процес оцінки і вибору може переслідувати декілька цілей, включаючи одну або більш з наступних:

- оцінка декількох CASE-засобів і вибір один або більш з них;
- оцінка одного або більш за CASE-засобів і збереження результатів для подальшого використання;
- вибір один або більш за CASE-засобів з використанням результатів попередніх оцінок.



Мал. 4.2. Модель процесу оцінки і вибору



Як видно з малюнка, вхідною інформацією для процесу оцінки є:

- визначення призначених для користувача потреб;
- цілі і обмеження проекту;
- дані про доступні CASE-засобів;
- список критеріїв, використовуваних в процесі оцінки.

Результати оцінки можуть включати результати попередніх оцінок. При цьому не слід забувати, що набір критеріїв, що використалися при попередній оцінці, повинен бути сумісним з поточним набором. Конкретний варіант реалізації процесу (оцінка і вибір, оцінка для майбутнього вибору або вибір, заснований на попередніх оцінках) визначається перерахованими вище цілями.

Елементи процесу включають:

- цілі, припущення і обмеження, які можуть уточнюватися в ході процесу;
- потреби користувачів, що відображають кількісні і якісні вимоги користувачів до CASE-засобів;
- критерії, що визначають набір параметрів, відповідно до яких проводиться оцінка і ухвалення рішення про вибір;
- формалізовані результати оцінок одного або більш за засоби;
- рішення, що рекомендується (зазвичай або рішення про вибір, або подальша оцінка).

Процес оцінки і/або вибору може бути початий тільки тоді, коли особа, група або організація повністю визначила для себе конкретні потреби і формалізувала їх у вигляді кількісних і якісних вимог в заданій наочній області. Термін "призначені для користувача вимоги" далі означає саме такі формалізовані вимоги.

Користувач повинен визначити конкретний порядок дій і ухвалення рішень з будь-якими необхідними ітераціями. Наприклад, процес може бути представлений у вигляді дерева рішень з його послідовним обходом і вибором підмножин кандидатів для детальнішої оцінки. Опис послідовності дій повинен визначати потік даних між ними.

Визначення списку критеріїв засноване на призначених для користувача вимогах і включає:

- вибір критеріїв для використання з приведенного далі переліку;
- визначення додаткових критеріїв;
- визначення області використання кожного критерію (оцінка, вибір або обидва процеси);
- визначення одній або більш за метрики для кожного критерію оцінки;
- призначення ваги кожному критерію при виборі.

#### **4.2.2. Процес оцінки**

Метою процесу оцінки є визначення функціональності і якості CASE-засобів для подальшого вибору. Оцінка виконується відповідно до конкретних критеріїв, її результати включають як об'єктивні, так і суб'єктивні

дані по кожному засобу.

Процес оцінки включає наступні дії:

- формулювання завдання оцінки, включаючи інформацію про мету і масштаби оцінки;
- визначення критеріїв оцінки, витікаюче з визначення завдання;
- визначення засобів-кандидатів шляхом проглядання списку кандидатів і аналізу інформації про конкретні засоби;
- оцінка засобів-кандидатів в контексті вибраних критеріїв. Необхідні для цього дані можуть бути отримані шляхом аналізу самих засобів і їх документації, опиту користувачів, роботи з демо-версіями, виконання тестових прикладів, експериментального застосування засобів і аналізу результатів попередніх оцінок;
- підготовка звіту за наслідками оцінки.

Одним з найважливіших критеріїв в процесі оцінки може бути потенційна можливість інтеграції кожного із засобів-кандидатів з іншими засобами, що вже знаходяться в експлуатації або планованими до використання в даній організації.

Масштаб оцінки повинен встановлювати необхідний рівень деталізації, необхідні ресурси і ступінь застосовності її результатів. Наприклад, оцінка повинна виконуватися для набору з одного або конкретніших CASE-засобів; CASE-засобів, що підтримують один або конкретніших процесів створення і супроводу ПЗ або CASE-засобів, що підтримують один або більш за проекти або типи проектів.

Список CASE-засобів - можливих кандидатів формується з різних джерел: оглядів ринку ПЗ, інформації постачальників, оглядів CASE-засобів і інших подібних публікацій.

Наступним кроком є отримання інформації про CASE-засоби або отримання їх самих або і те, і інше. Ця інформація може складатися з оцінок незалежних експертів, повідомлень і звітів постачальників CASE-засобів, результатів демонстрації можливостей CASE-засобів з боку постачальників і інформації, отриманої безпосередньо від реальних користувачів. Самі CASE-засоби можуть бути отримані шляхом придбання, у вигляді оцінної копії або іншими способами.

Оцінка і накопичення відповідних даних може виконуватися наступними способами:

- аналіз CASE-засобів і документації постачальника;
- опит реальних користувачів;
- аналіз результатів проектів, що використали дані CASE-засоби;
- проглядання демонстрацій і опит демонстраторів;
- виконання тестових прикладів;
- застосування CASE-засобів в пілотних проектах;
- аналіз будь-яких доступних результатів попередніх оцінок.

Існують як об'єктивні, так і суб'єктивні критерії. Результати оцінки відповідно до конкретного критерію можуть бути двійковими, знаходитися в

деякому числовому діапазоні, бути просто числовим значенням або мати яку-небудь іншу форму.

Для об'єктивних критеріїв оцінка повинна виконуватися шляхом відтворної процедури, щоб будь-який інший фахівець, що виконує оцінку, міг отримати такі ж результати. Якщо використовуються тестові приклади, їх набір повинен бути заздалегідь визначений, уніфікований і документований.

По суб'єктивних критеріях CASE-засобів повинен оцінюватися групою фахівців, що використовують одні і ті ж критерії. Необхідний рівень досвіду фахівців або груп повинен бути заздалегідь визначений.

Результати оцінки повинні бути стандартним чином документовані (для полегшення подальшого використання) і, при необхідності, затверджені.

Звіт за наслідками оцінки повинен містити наступну інформацію:

- введення. Загальний огляд процесу і перелік основних результатів;
- передумови. Мета оцінки і бажані результати, період часу, протягом якого виконувалася оцінка, визначення ролей і відповідного досвіду фахівців, що виконували оцінку;

- підхід до оцінки. Опис загального підходу, включаючи отримані CASE-засоби, інформацію, що визначає контекст і масштаб оцінки, а також будь-які припущення і обмеження;

- інформація про CASE-засобах. Вона повинна включати наступне: 1) найменування CASE-засобів; 2) версію CASE-засобів; 3) дані про постачальника, включаючи контактну адресу і телефон; 4) конфігурацію технічних засобів; 5) вартісні дані; 6) опис CASE-засобів, що включає підтримувані даним засобом процеси створення і супроводу ПЗ, програмне середовище CASE-засобів (зокрема, підтримувані мови програмування, операційні системи, сумісність з базами даних), функції CASE-засобів, входные/выходные дані і область застосування;

- етапи оцінки. Конкретні дії, що виконуються в процесі оцінки, повинні бути описані із ступенем деталізації, необхідної як для розуміння масштабу і глибини оцінки, так і для її повторення при необхідності;

- конкретні результати. Результати оцінки повинні бути представлені в термінах критеріїв оцінки. У тих випадках, коли звіт охоплює цілий ряд CASE-засобів або результати даної оцінки зіставлятимуться з аналогічними результатами інших оцінок, необхідно звернути особливу увагу на формат представлення результатів, сприяючий такому порівнянню. Суб'єктивні результати повинні бути відокремлені від об'єктивних і повинні супроводжуватися необхідними поясненнями;

- виводи і висновки;

- застосування. Формулювання завдання оцінки і уточнений список критеріїв.

#### **4.2.3. Процес вибору**

Процеси оцінки і вибору тісно взаємозв'язані один з одним. За наслідками оцінки мети вибору і/або критерії вибору і їх ваги можуть зажадати модифікації. У таких випадках може потрібно повторна оцінка.

Коли аналізуються остаточні результати оцінки і до них застосовуються критерії вибору, може бути рекомендоване придбання CASE-засобів або набору CASE-засобів. Альтернативою може бути відсутність адекватних CASE-засобів, в цьому випадку рекомендується розробити новий CASE-засобів, модифікувати що існує або відмовитися від впровадження.

Процес вибору тісно взаємозв'язаний з процесом оцінки і включає наступні дії:

- формулювання завдань вибору, включаючи цілі, припущення і обмеження;

- виконання всіх необхідних дій з вибору, включаючи визначення і ранжирування критеріїв, визначення засобів-кандидатів, збір необхідних даних і застосування ранжированих критеріїв до результатів оцінки для визначення засобів з якнайкращими показниками. Для багатьох користувачів важливим критерієм вибору є інтегрованість CASE-засобів з існуючим середовищем;

- виконання необхідної кількості ітерацій з тим, щоб вибрати (або відкинути) засоби, що мають схожі показники;

- підготовка звіту за наслідками вибору.

В процесі вибору можливе отримання двох результатів:

- рекомендацій по вибору конкретного CASE-засобів;

- запиту на отримання додаткової інформації до процесу оцінки.

Масштаб вибору повинен встановлювати необхідний рівень деталізації, необхідні ресурси, графік і очікувані результати. Існує ряд параметрів, які можуть бути використані для визначення масштабу, включаючи:

- використання попереднього відбору (наприклад, відбір тільки засобів, що працюють на конкретній платформі);

- використання раніше отриманих результатів оцінки, результатів оцінки із зовнішніх джерел або комбінації того і іншого;

В тому випадку, якщо попередні оцінки виконувалися з використанням різних наборів критеріїв або виконувалися з використанням конкретних критеріїв, але різними способами, результати оцінок повинні бути представлені в узгодженій формі. Після завершення даного кроку оцінка кожного CASE-засобів повинна бути представлена в рамках єдиного набору критеріїв і повинна бути безпосередньо сопоставима з іншими оцінками.

Алгоритми, зазвичай використовувані для вибору, можуть бути засновані на масштабі або ранзі. Алгоритми, засновані на масштабі, обчислюють єдине значення для кожного CASE-засобів шляхом множення ваги кожного критерію на його значення (з урахуванням масштабу) і складання всіх творів. CASE-засобів з найвищим результатом отримує перший ранг. Алгоритми, засновані на ранзі, використовують ранжирування CASE-засобів - кандидатів по окремих критеріях або групах критеріїв відповідно до значень критеріїв в заданому масштабі. Потім, аналогічно попередньому, ранги зводяться разом і обчислюються загальні значення рангів.

При аналізі результатів вибору передбачається, що процес вибору завершений, CASE-засобів вибраний і рекомендований до використання. Проте, може потрібно точніший аналіз для визначення ступеня залежності значень ключових критеріїв від відмінностей в значеннях характеристик CASE-засобів - кандидатів. Такий аналіз дозволить визначити, наскільки результат ранжирування CASE-засобів залежить від оптимальності вибору вагових коефіцієнтів критеріїв. Він також може використовуватися для визначення істотних відмінностей між CASE-засобами з дуже близькими значеннями критеріїв або рангами.

Якщо жодне з CASE-засобів не задовольняє мінімальним критеріям, вибір (можливо, разом з оцінкою) може бути повторений для інших CASE-засобів - кандидатів.

Якщо відмінності між найпереважнішими кандидатами неістотні, додаткова інформація може бути отримана шляхом повторного вибору (можливо, разом з оцінкою) з використанням додаткових або інших критеріїв.

Рекомендації по вибору повинні бути строго обгрунтовані. У разі відсутності адекватних CASE-засобів, як було відмічено вищим, рекомендується розробити новий CASE-засобів, модифікувати що існує або відмовитися від впровадження.

#### **4.2.4. Критерії оцінки і вибору**

Критерії формують базис для процесів оцінки і вибору і можуть приймати різні форми, включаючи:

- числові заходи в широкому діапазоні значень, наприклад, об'єм необхідної пам'яті;
- числові заходи в обмеженому діапазоні значень, наприклад, простота освоєння, виражена в балах від 1 до 5;
- двійкові заходи (істина/брехня, да/нет), наприклад, здатність генерації документації у форматі Postscript;
- заходи, які можуть приймати одне або більш з кінцевої безлічі значень, наприклад, платформи, для яких підтримується CASE-засобів.

Типовий процес оцінки і/або вибору може використовувати набір критеріїв різних типів.

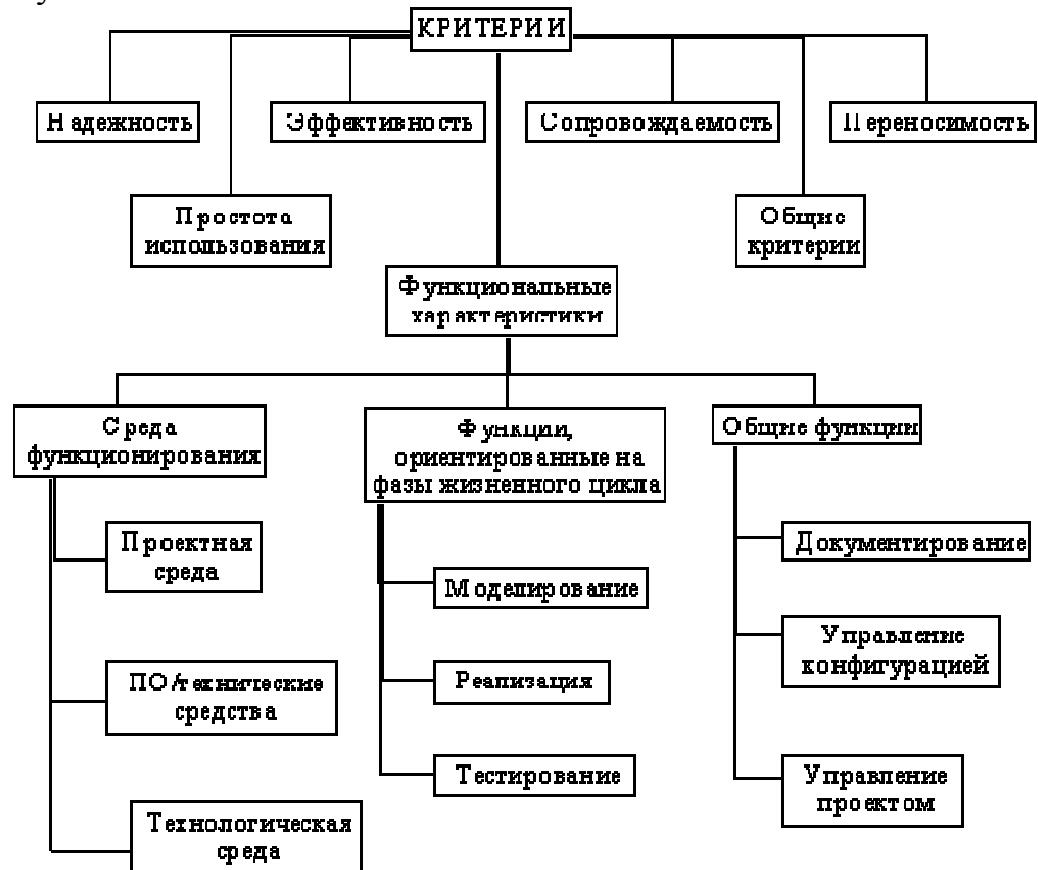
Структура набору критеріїв приведена на малюнку 4.3. Кожен критерій повинен бути вибраний і адаптований експертом з урахуванням особливостей конкретного процесу. В більшості випадків тільки деякі з безлічі описаних нижче критеріїв виявляються прийнятними для використання, при цьому також додаються додаткові критерії. Вибір і уточнення набору використовуваних критеріїв є критичним кроком в процесі оцінки і/або вибору.

#### *Функціональні характеристики*

Критерії першого класу призначені для визначення функціональних характеристик CASE-засобів. Вони у свою чергу підрозділяються на ряд груп і підгруп.

1. Середовище функціонування:

- а. Проектне середовище:
- підтримка процесів життєвого циклу. Визначає набір процесів ЖЦ, які підтримує CASE-засобівом. Прикладами таких процесів є аналіз вимог, проектування, реалізація, тестування і оцінка, супровід, забезпечення якості, управління конфігурацією і управління проектом, причому вони залежать від прийнятої користувачем моделі ЖЦ.
  - область застосування. Прикладами є системи обробки транзакцій, системи реального часу, інформаційні системи і так далі
  - розмір підтримуваних застосувань. Визначає обмеження на такі величини, як кількість рядків коду, рівнів вкладеності, розмір бази даних, кількість елементів даних, кількість об'єктів конфігураційного управління.
- б. ПО/ТЕХНІЧЕСЬКІЕ засобу:
- необхідні технічні засоби. Устаткування, необхідне для функціонування CASE-засобівом, включаючи тип процесора, об'єм оперативної і дискової пам'яті.
  - підтримувані технічні засоби. Елементи устаткування, які можуть використовуватися CASE-засобівом, наприклад, пристрою введення/виводу.
  - потрібне ПЗ. ПЗ, необхідне для функціонування CASE-засобівом, включаючи операційні системи і графічні оболонки.
  - підтримуване ПЗ. Програмні продукти, які можуть використовуватися CASE-засобівом.



Мал. 4.3. Структура набора критеріів

- с. Технологічне середовище:
- відповідність стандартам технологічного середовища. Такі

стандарти стосуються мови, бази даних, репозиторія, комунікацій, графічного інтерфейсу користувача, документації, розробки, управління конфігурацією, безпеки, стандартів обміну інформацією і інтеграції по даним, по управлінню і по призначеному для користувача інтерфейсу.

- сумісність з іншими засобами. Здібність до взаємодії з іншими засобами, включаючи безпосередній обмін даними (прикладом таких засобів є текстові процесори, бази даних і інші CASE-засоби). Можливість перетворення репозиторія або його частини в стандартний формат для обробки іншими засобами.

- підтримувана методологія. Набір методів і методик, підтримуваних CASE-засобом. Прикладами є структурний або об'єктно-орієнтований аналіз і проектування.

- підтримувані мови. Всі мови, використовувані CASE-засобом. Прикладами таких мов є мови програмування (Кобол, Ада, 3), мови баз даних і мови запитів (DDL, SQL), графічні мови (Postscript, HPGL), мови специфікації проектних вимог і інтерфейси операційних систем (мови управління завданнями).

## 2. Функції, орієнтовані на фази життєвого циклу:

### а. Моделювання:

Дані критерії визначають здатність виконання функцій, необхідних для специфікації вимог до ПЗ і перетворенню їх в проект:

- побудова діаграм. Можливість створення і редагування діаграм різних типів, що представляють інтерес для користувача. Найбільш поширені типи діаграм описані в розділі 2.

- графічний аналіз. Можливість аналізу графічних об'єктів, а також зберігання і представлення проектної інформації в графічному уявленні. В більшості випадків графічні аналізатори інтегровані із засобами побудови діаграм.

- введення і редагування специфікацій вимог і проектних специфікацій. До специфікацій такого роду відносяться описи функцій, даних, інтерфейсів, структури, якості, продуктивності, технічних засобів, середовища, витрат і графіків.

- мова специфікації вимог і проектних специфікацій. Можливість імпорту, експорту і редагування специфікацій з використанням формальної мови.

- моделювання даних. Можливість введення і редагування інформації, що описує елементи даних системи і їх відношення.

- моделювання процесів. Можливість введення і редагування інформації, що описує процеси системи і їх відношення.

- проектування архітектури ПЗ. Проектування логічної структури ПЗ - структури модулів, інтерфейсів і ін.

- імітаційне моделювання. Можливість динамічного моделювання різних аспектів функціонування системи на основі специфікацій вимог і/або проектних специфікацій, включаючи зовнішній інтерфейс і продуктивність

(наприклад, час відгуку, коефіцієнт використання ресурсів і пропускну спроможність).

- прототипирование. Можливість проектування і генерації попереднього варіанту всієї системи або її окремих компонент на основі специфікацій вимог і/або проектних специфікацій. Прототіпірованіє в основному стосується зовнішнього призначеного для користувача інтерфейсу і здійснюється при безпосередній участі користувачів.

- генерація екранних форм. Можливість генерації екранних форм на основі специфікацій вимог і/або проектних специфікацій.

- можливість трасування. Можливість крізного аналізу функціонування системи від специфікації вимог до кінцевих результатів (встановлення і відстежування відповідностей і зв'язків між функціональними і іншими зовнішніми вимогами до ІС, технічними рішеннями і результатами проектування). Пряме трасування (перевірка обліку всіх вимог) і зворотне трасування (пошук проектних рішень, не пов'язаних ні з якими зовнішніми вимогами).

- синтаксичний і семантичний контроль проектних специфікацій. Контроль синтаксису діаграм і типів їх елементів, контроль декомпозиції функцій, перевірка специфікацій на повноту і несуперечність.

- інші види аналізу. Конкретні додаткові види аналізу можуть включати алгоритми, потоки даних, нормалізацію даних, використання даних, призначений для користувача інтерфейс.

- автоматизоване проектування звітів.

#### b. Реалізація:

Реалізація зачіпає функції, пов'язані із створенням виконуваних елементів системи (програмних код) або модифікацією існуючої системи. Багато хто з перерахованих нижче критеріїв залежить від конкретних мов і включають наступні:

- синтаксично кероване редагування. Можливість введення і редагування початкових код на одному або декількох мовах з одночасним синтаксичним контролем.

- генерація коду. Можливість генерації код на одному або декількох мовах на основі проектних специфікацій. Типи коду, що генерується, можуть включати звичайний програмний код, схему бази даних, запити, екрани/меню.

- компіляція коду.

- конвертація початкової коду. Можливість перетворення коду з однієї мови в іншій.

- аналіз надійності. Можливість кількісно оцінювати параметри надійності ПЗ, такі, як кількість помилок і ін.

- реверсний інжиніринг. Можливість аналізу існуючих початкових код і формування на їх основі проектних специфікацій.

- реструктуризація початкової коду. Можливість модифікації формату і/або структури існуючої початкової коду.



– аналіз початкової коди. Прикладами такого аналізу можуть бути визначення розміру коди, обчислення показників складності, генерація перехресних посилань і перевірка на відповідність стандартам.

– відладка. Типові функції відладки - трасування програм, виділення вузьких місць і найбільш часто використовуваних фрагментів коди і так далі

с. Тестування:

Критерії тестування включають наступні:

– опис тестів. Типові можливості включають генерацію тестових даних, алгоритмів тестування, необхідних результатів і так далі

– фіксація і повторення дій оператора. Можливість фіксувати дані, що вводяться оператором за допомогою клавіатури, миші і так далі, редагувати їх і відтворювати в тестових прикладах.

– автоматичний запуск тестових прикладів.

– регресійне тестування. Можливість повторення і модифікації раніше виконаних тестів для визначення відмінностей в системі і/або середовищі.

– автоматизований аналіз результатів тестування. Типові можливості включають порівняння очікуваних і реальних результатів, порівняння файлів, статистичний аналіз результатів і ін.

– аналіз тестового покриття. Оснащеність засобами контролю початкової коди і аналіз тестового покриття. Перевіряються, зокрема, звернення до операторів, процедур і змінних.

– аналіз продуктивності. Можливість аналізу продуктивності програм. Аналізовані параметри продуктивності можуть включати використання центрального процесора, пам'яті, звернення до певних елементів даних і/або сегментам коди, тимчасові характеристики і так далі

– аналіз виняткових ситуацій в процесі тестування.

– динамічне моделювання середовища. Зокрема, можливість автоматично генерувати модельовані вхідні дані системи.

3. Загальні функції:

Приведені нижче критерії визначають функції CASE-засобів, що охоплюють всю сукупність фаз ЖЦ. Підтримка всіх цих функцій здійснюється за допомогою репозиторія.

а. Документування:

– редагування текстів і графіки. Можливість вводити і редагувати дані в текстовому і графічному форматі.

– редагування за допомогою форм. Можливість підтримувати форми, визначені користувачами, вводити і редагувати дані відповідно до форм.

– можливості видавничих систем.

– підтримка функцій і форматів гіпертексту.

– відповідність стандартам документування.

– автоматичне витягання даних з репозиторія і генерація документації по специфікаціях користувача.

б. Управління конфігурацією:

– контроль доступу і змін. Можливість контролю доступу на

фізичному рівні до елементів даних і контролю змін. Контроль доступу включає можливості визначення прав доступу до компонентів, а також витягання елементів даних для модифікації, блокування доступу до них на час модифікації і приміщення назад в репозиторій.

- відстежування модифікацій. Фіксація і ведення журналу всіх модифікацій, внесених до системи в процесі розробки або супроводу.

- управління версіями. Ведення і контроль даних про версії системи і всіх її колективно використовуваних компонентах.

- облік стану об'єктів конфігураційного управління. Можливість отримання звітів про всі послідовні версії, вміст і стан різних об'єктів конфігураційного управління.

- генерація версій і модифікацій. Підтримка призначеного для користувача опису послідовності дій, потрібних для формування версій і модифікацій, і автоматичне виконання цих дій.

- архівація. Можливість автоматичної архівації елементів даних для подальшого використання.

с.      Управління проектом:

- управління роботами і ресурсами. Контроль і управління процесом проектування ІС в термінах структури завдань і призначення виконавців, послідовності їх виконання, завершеності окремих етапів проекту і проекту в цілому. Можливість підтримки планових даних, фактичних даних і їх аналізу. Типові дані включають графіки (з урахуванням календаря, робочого годинника, вихідних і ін.), комп'ютерні ресурси, розподіл персоналу, бюджет і ін.

- оцінка. Можливість оцінювати витрати, графік і інші проектні параметри, що вводяться користувачами.

- управління процедурою тестування. Підтримка управління процедурами і програмою тестування, наприклад, управління розкладом планованих процедур, фіксація і запис результатів тестування, генерація звітів і так далі

- управління якістю. Введення відповідних даних, їх аналіз і генерація звітів.

- дії, що коректують. Підтримка управління діями, що коректують, включаючи обробку повідомлень про проблемні ситуації.

#### **4.2.4.1. Надійність**

- адміністрування репозиторія. Контроль і забезпечення цілісності проектних даних.

- автоматичне резервування (визначуване постачальником або плановане користувачем).

- безпека. Захист від несанкціонованого доступу.

- обробка помилок. Виявлення помилок в роботі системи, сповіщення користувача, коректне завершення роботи або збереження стану до моменту переривання.

- аналіз відмов в критичних застосуваннях.

#### **4.2.4.2. Простота використання**

– зручність призначеного для користувача інтерфейсу. Зручність розташування і представлення часто використовуваних елементів екрану, способів введення даних і ін.

– локалізація (відповідно до вимог даної країни).

– простота освоєння. Трудові і тимчасові витрати на освоєння засобів.

– адаптується до конкретних вимог користувача. Адаптується до різних алфавітів, режимів текстового і графічного уявлення (зліва-направо, зверху-вниз), різних форматів дати, способів введення/виводу (екранним формам і форматам), змін в методології (змінам графічних нотацій, правил, властивостей і складу зумовлених об'єктів) і ін.

– якість документації (повнота, зрозумілість, легкість для читання, корисність і ін.).

– доступність і якість учбових матеріалів. Вони можуть включати комп'ютерні учбові матеріали, навчальні посібники, курси.

– вимоги до рівня знань. Кваліфікація і досвід, необхідні для ефективного використання CASE-засобів.

– простота роботи з CASE-засобівом (як для початківців, так і для досвідчених користувачів).

– уніфікована призначеного для користувача інтерфейсу (по відношенню до інших засобів, що використовуються в даній організації).

– онлайніві підказки (повнота і якість).

– якість діагностики (зрозумілість і корисність діагностичних повідомлень для користувача).

– допустимий час реакції на дії користувача (залежно від середовища).

– простота установки і оновлення версій.

#### **4.2.4.3. Ефективність**

– вимоги до технічних засобів. Вимоги до оптимального розміру зовнішньої і оперативної пам'яті, типу і продуктивності процесора, що забезпечує прийнятний рівень продуктивності.

– ефективність робочого навантаження. Ефективність виконання CASE-засобівом своїх функцій залежно від інтенсивності роботи користувача (наприклад, кількість натиснень клавіш або кнопки миші, потрібне для виконання певних функцій).

– продуктивність. Час, що витрачається CASE-засобівом для виконання конкретних завдань (наприклад, час відповіді на запит, час аналізу 100000 рядків коду). В деяких випадках дані оцінки продуктивності можна отримати із зовнішніх джерел.

#### **4.2.4.4. Сопроводжаємость**

– рівень підтримки з боку постачальника (швидкість дозволу проблем, постачання нових версій, забезпечення додаткових можливостей).

– трасованість оновлень (простота освоєння відмінностей нових версій від тих, що існують).

- сумісність оновлень (сумісність нових версій з тими, що існують, включаючи, наприклад, сумісність за вхідними або вихідними даними).
- супроводжує кінцевого продукту (простота внесення змін в ПЗ і документацію).

#### **4.2.4.5. Переносимість**

- сумісність з версіями ОС (можливість роботи в середовищі різних версій однієї і тієї ж ОС, простота модифікації CASE-засобів для роботи з новими версіями ОС).
- переносимість даних між різними версіями CASE-засобів.
- відповідність стандартам переносимості. Такі стандарти включають документацію, комунікації і призначений для користувача інтерфейс, віконний інтерфейс, мови програмування, мови запитів і ін.

#### **4.2.4.6. Загальні критерії**

Приведені нижче критерії є загальними за своєю природою і не належать до сукупності показників якості, приведеної в стандарті ISO/IEC 9126: 1991.

- витрати на CASE-засобів. Включають вартість придбання, установки, початкового супроводу і навчання. Слід враховувати ціну для всіх необхідних конфігурацій (включаючи єдину копію, декілька копій, локальну ліцензію, ліцензію для підприємства, мережеву ліцензію).

- оцінний ефект від впровадження CASE-засобів (рівень продуктивності, якості і так далі). Така оцінка може зажадати економічного аналізу.

- профіль дистриб'ютора. Загальні показники можливостей дистриб'ютора. Профіль дистриб'ютора може включати величину його організації, стаж в бізнесі, фінансове положення, список будь-яких додаткових продуктів, ділові зв'язки (зокрема, з іншими дистриб'юторами даного засобу), планована стратегія розвитку.

- сертифікація постачальника. Сертифікати, отримані від спеціалізованих організацій в області створення ПЗ (наприклад, SEI і ISO), засвідчують, що кваліфікація постачальника в області створення і супроводу ПЗ задовольняє деяким мінімально необхідним або цілком певним вимогам. Сертифікація може бути неформальною, наприклад, на основі аналізу якості роботи постачальника.

- ліцензійна політика. Доступні можливості ліцензування, право копіювання (носіїв і документації), будь-які обмеження і/або штрафні санкції за вторинне використання (мається на увазі продаж користувачем CASE-засобів продуктів, до складу яких входять деякі компоненти CASE-засобів, що використалися при розробці продуктів).

- експортні обмеження.

- профіль продукту. Загальна інформація про продукт, включаючи термін його існування, кількість проданих копій, наявність, розмір і рівень діяльності призначеної для користувача групи, система звітів про проблеми, програма розвитку продукту, сукупність застосувань, наявність помилок і ін.

– підтримка постачальника. Доступність, реактивність і якість послуг, що надаються постачальником для користувачів CASE-засобів. Такі послуги можуть включати телефонну "гарячу лінію", місцеву технічну підтримку, підтримку в самій організації.

– доступність і якість навчання. Навчання може проводитися на території постачальника, користувача або де-небудь у іншому місці.

– адаптація, потрібна для впровадження CASE-засобів в організації користувача. Прикладом може бути визначення способу використання централізованого CASE-засобів з єдиною, загальною БД в розподіленому середовищі.

#### **4.2.5. Приклад підходу до визначення критеріїв вибору CASE-засобів**

Приклад підходу до визначення критеріїв вибору CASE-засобів приведений в [1]. Передбачається, що CASE-засоби будуть використані в крупному типовому проекті ІС, що володіє характеристиками, перерахованими у введенні. У загальному випадку стратегія вибору CASE-засобів для конкретного застосування залежить від цілей, потреб і обмежень майбутнього проекту ІС (включаючи кваліфікацію фахівців, що беруть участь в процесі проектування), які, у свою чергу, визначають використовувану методологію проектування.

Слід підкреслити, що визначальним чинником при вибиранні тих або інших інструментальних засобів є використовувана методологія і технологія проектування, а не навпаки. З цієї точки зору безглуздо порівнювати CASE-засобів самі по собі у відриві від методології, оскільки ІС можна в принципі розробити будь-якими засобами.

Традиційно при обговоренні проблеми вибору CASE-засобів велика увага приділялася особливостям реалізації тієї або іншої методології аналізу наочної області (E-R, IDEF0, IDEF1X, Gane/Sarson, Yourdon, Barker і ін.). Безумовно, багатство образотворчих і описових засобів дає можливість на етапах стратегічного планування і аналізу побудувати якнайповнішу і адекватнішу модель наочної області. З іншого боку, якщо говорити про кінцеві результати - бази даних і застосування, то виявляється, що частина описів в них практично не відбивається, залишаючись чисто декларативною (на виході ми у будь-якому випадку отримуємо опис БД в табличному уявленні з мінімальним набором обмежень цілісності і здійснимий код застосувань, велику частину яких складають екранні форми, що не виводяться безпосередньо з моделей наочної області). Досвідчені аналітики і проектувальники завжди з великими або меншими трудовитратами прийдуть до потрібного кінцевого результату незалежно від того, яка конкретно методологія або її різновид реалізована в даному інструменті. Це, звичайно, не означає, що методологія не важлива, навпаки, відсутність або неповнота описових засобів можуть з самого початку значного утруднити роботу над проектом. Проте, часто на першому плані виявляються інші критерії, невиконання яких може породити набагато більші труднощі.

Як було відмічено в підрозділі 1.3, технологія проектування повинна

бути підтримана комплексом узгоджених CASE-засобів, що забезпечують автоматизацію процесів, що виконуються на всіх стадіях ЖЦ. Може створитися враження, що якщо можна сформувати необхідну апаратну платформу з компонентів різних фірм-виробників, то так само просто можна вибрати і скомплексувати різні інструментальні засоби, кожне з яких є одним зі світових лідерів в своєму класі. Проте для інструментальних засобів в даний час, на відміну від устаткування, відсутні міжнародні стандарти на основні властивості кінцевих продуктів (програм, баз даних і їх сполучення). Оскільки складові частини проекту повинні бути інтегровані в єдиний продукт, отже, має сенс розглядати не будь-які, а тільки зв'язані інструментальні засоби, які в принципі можуть бути орієнтовані, - навіть усередині одного класу - на різні методології; при цьому необхідно відбирати до складу комплексу CASE-засобів засобу, близькі методології, що підтримують принаймні, якщо не одну і ту ж.

Виходячи з перерахованих вище міркувань, як основні критерії вибору CASE-засобів приймаються наступні критерії:

1. Підтримка повного життєвого циклу ІС із забезпеченням еволюційності її розвитку

Повний життєвий цикл ІС повинен підтримуватися комплексом інструментальних засобів, перерахованих в розділі 3.2, з урахуванням наступних особливостей:

- колективного характеру розробки моделей, проектних специфікацій і застосувань територіально розподіленими групами фахівців з використанням різних інструментальних засобів (включаючи їх інтеграцію, тестування і відладку);

- необхідності адаптації типового проекту до різних системно-технічних платформ (технічним засобам, операційним системам і СУБД) і організаційно-економічних особливостей об'єктів впровадження;

- необхідності інтеграції з існуючими розробками (включаючи реінжиніринг застосувань і конвертація БД). Для тих, що існують ІС повинен забезпечуватися плавний перехід із старого середовища експлуатації в нову з мінімальними переробками і підтримкою експлуатованих баз даних і застосувань, упроваджених до початку робіт із створення нової системи.

2. Забезпечення цілісності проекту і контролю за його станом

Дану вимогу означає наявність єдиного технологічного середовища створення, супроводу і розвитку ІС, а також цілісність репозиторія. Єдине технологічне середовище повинне забезпечуватися за рахунок використання єдиного CASE-засобів для підтримки моделей ІС, а також за рахунок наявності програмно-технологічних інтерфейсів між окремими інструментальними засобами, сертифікованих і підтримуваних фірмами-розробниками відповідних засобів. Зокрема, інтерфейс між CASE-засобівими і засобами розробки застосувань повинен виконувати дві основні функції: а) безпосередній перехід в рамках єдиного середовища від опису логіки застосування, реалізованого CASE-засобівом, до розробки призначеного для користувача інтерфейсу (екранних форм); б) перенесення опису БД з

репозиторія CASE-засобів в репозиторій засобу розробки застосувань і назад. Вся інформація про проект повинна автоматично поміщатися в базу проектних даних, при цьому повинні підтримуватися узгодженість, несуперечність, повнота і мінімальна надмірність проекту, а також коректність операцій його редагування. Це може бути досягнуто за умови виключення або істотного обмеження можливості актуалізації репозиторія різними засобами. В рамках CASE-засобів повинні забезпечуватися контроль відповідності декомпозицій діаграм, а також контроль відповідності діаграм різних типів (наприклад, діаграм потоків даних і ER-діаграмм).

Невиконання вимоги цілісності в умовах роз'єднаності розробників і тимчасової протяжності крупного проекту може означати втрату контролю за його станом.

### 3. Незалежність від програмно-апаратної платформи і СУБД

Вимога визначається неоднорідністю середовища функціонування ІС. Така незалежність може мати дві складових: незалежність середовища розробки і незалежність середовища експлуатації застосувань. Вона забезпечується за рахунок наявності сумісних версій CASE-засобів для різних платформ і драйверів відповідних мережевих протоколів, менеджерів транзакцій і СУБД.

### 4. Підтримка одночасної роботи груп розробників

Розвинені CASE-засобів повинні володіти можливостями розділення повноважень персоналу розробників і об'єднання окремих робіт в загальний проект. Повинна забезпечуватися одночасна (у заданій мережевій конфігурації) робота проектувальників БД і розробників застосувань (розробники застосувань в такій ситуації можуть починати роботу з базою даних, не чекаючи повного завершення її проектування CASE-засобами). При цьому всі групи фахівців повинні бути забезпечені адекватним інструментарієм, а внесення змін до проекту різними розробниками повинно бути узгодженим і коректним. Кожен розробник повинен мати можливість роботи зі своїм особистим репозиторієм, фрагментом, що є, або копією загального репозиторія. Повинні забезпечуватися змістовна інтеграція всіх змін, що вносяться розробниками, в загальному репозиторії, одночасна доступність для розробника загального і особистого репозиторіїв і простота перенесення об'єктів між ними.

### 5. Можливість розробки застосувань "клієнт-сервер" необхідної конфігурації

Мається на увазі поєднання наявності розвиненого графічного середовища розробки застосувань (многооконність, різноманітність стандартних графічних об'єктів, різноманітність використовуваних шрифтів і так далі) з можливістю декомпозиції (partitioning) застосування на "клієнтську" частину, що реалізовує призначений для користувача екранний інтерфейс і "серверну" частину. При цьому повинна забезпечуватися можливість переміщення окремих компонентів застосування між "клієнтом" і "сервером" на найбільш відповідну платформу, що забезпечує максимальну

ефективність функціонування застосування в цілому, а також можливість використання сервера застосувань (менеджера транзакцій).

#### 6. Відкрита архітектура і можливості експорту/імпорту

Відкрита і загальнодоступна інформація про використовувані формати даних і прикладних програмних інтерфейсах повинна дозволяти інтегрувати інструментальні засоби третіх фірм і відносно безболісно переходити від однієї системи до іншої. Можливості експорту/імпорту означають, що специфікації, отримані на етапах аналізу, проектування і реалізації для однієї ІС, можуть бути використані для проектування іншої ІС. Повторне проектування і реалізація можуть бути забезпечені за допомогою засобів експорту/імпорту специфікацій в різні CASE-засобів.

#### 7. Якість технічної підтримки в Росії, вартість придбання і підтримки, досвід успішного використання

Мається на увазі наявність кваліфікованих дистриб'юторів і консультантів, швидкість обслуговування користувачів, висока якість технічної підтримки і навчання продукту і методології його застосування для великих колективів розробників (наявність відомостей про практику використання системи, якість документації, укомплектованість прикладами і повчальними курсами, наявність пілотних проектів). Витрати на навчання новим технологіям значительны, проте втрати від використання сучасних складних технологій ненавченими фахівцями можуть виявитися значно вище.

Крім того, фірми-постачальники інструментальних засобів повинні бути стійкими, оскільки технологія вибирається не на один рік, а також повинні забезпечувати хорошу підтримку на території Росії (гаряча лінія, консультації, навчання, консалтинг), можливо, через дистриб'юторів.

Що стосується вартості, слід враховувати можливість отримання безкоштовної тимчасової ліцензії, вартість ліцензії на одне робоче місце CASE-засобів, знижки, що надаються фірмою у разі придбання великої кількості ліцензій, необхідність придбання run-time версій для експлуатації застосувань і так далі. В той же час вартість продукту повинна розглядатися не сама по собі, а з урахуванням її відповідності можливостям продукту.

#### 8. Простота освоєння і використання

Враховуються наступні характеристики:

- відповідність інструменту особливостям і потенційним можливостям колективу розробників;
- доступність призначеного для користувача інтерфейсу;
- час, необхідний для навчання;
- простота установки;
- якість документації;
- об'єм ручної праці при супроводі ІС.

#### 9. Забезпечення якості проектної документації

Ця вимога відноситься до можливостей CASE-засобів аналізувати і перевіряти описи і документацію на повноту і несуперечність, а також на відповідність прийнятим в даній методології стандартам і правилам



(включаючи ГОСТ, ЕСПД). В результаті аналізу повинна формуватися інформація, вказуюча на наявні суперечності або неповноту в проектній документації. Повинна бути також забезпечена можливість створювати нові форми документів, визначувані користувачами.

10. Використання загальноприйнятих, стандартних нотацій і угод

Для того, щоб проект міг виконуватися різними колективами розробників, необхідне використання стандартних методів моделювання і стандартних нотацій, які повинні бути оформлені у вигляді нормативів до початку процесу проектування. Недотримання проектних стандартів ставить розробників в залежність від фірми-виробника даного засобу, робить скрутним формальний контроль коректності проектних рішень і знижує можливості залучення додаткових колективів розробників, зміни виконавців і відчуження проекту, оскільки число фахівців, знайомих з даним методом (нотацією) може бути обмеженим.

В результаті виконаного аналізу може опинитися, що жоден доступний засіб не задовольняє в потрібній мірі всім основним критеріям і не покриває всі потреби проекту. В цьому випадку може застосовуватися набір засобів, що дозволяє побудувати на їх базі єдине технологічне середовище.

#### **4.3. Виконання пілотного проекту**

Перед повномасштабним впровадженням вибраного CASE-засобу в організації виконується пілотний проект, метою якого є експериментальна перевірка правильності рішень, прийнятих на попередніх етапах, і підготовка до впровадження.

Пілотним проектом є первинне реальне використання CASE-засобу в призначеному для цього середовищі і зазвичай має на увазі ширший масштаб використання CASE-засобу по відношенню до того, який був досягнутий під час оцінки. Пілотний проект повинен володіти багатьма з характеристик реальних проектів, для яких призначений даний засіб. Він переслідує наступні цілі:

- підтвердити достовірність результатів оцінки і вибору;
- визначити, чи даний CASE-засіб годиться для використання в даній організації, і якщо так, то визначити найбільш відповідну область його застосування;
- зібрати інформацію, необхідну для розробки плану практичного впровадження;
- набути власного досвіду використання CASE-засобу.

Пілотний проект дозволяє отримати важливу інформацію, необхідну для оцінки якості функціонування CASE-засобу і його підтримки з боку постачальника після того, як засіб встановлений.

Важливою функцією пілотного проекту є ухвалення рішення щодо придбання або відмови від використання CASE-засобу. Провал пілотного проекту дозволяє уникнути значніших і дорожчих невдач надалі, оскільки пілотний проект зазвичай пов'язаний з придбанням відносно невеликої кількості ліцензій і навчанням вузького круга фахівців.

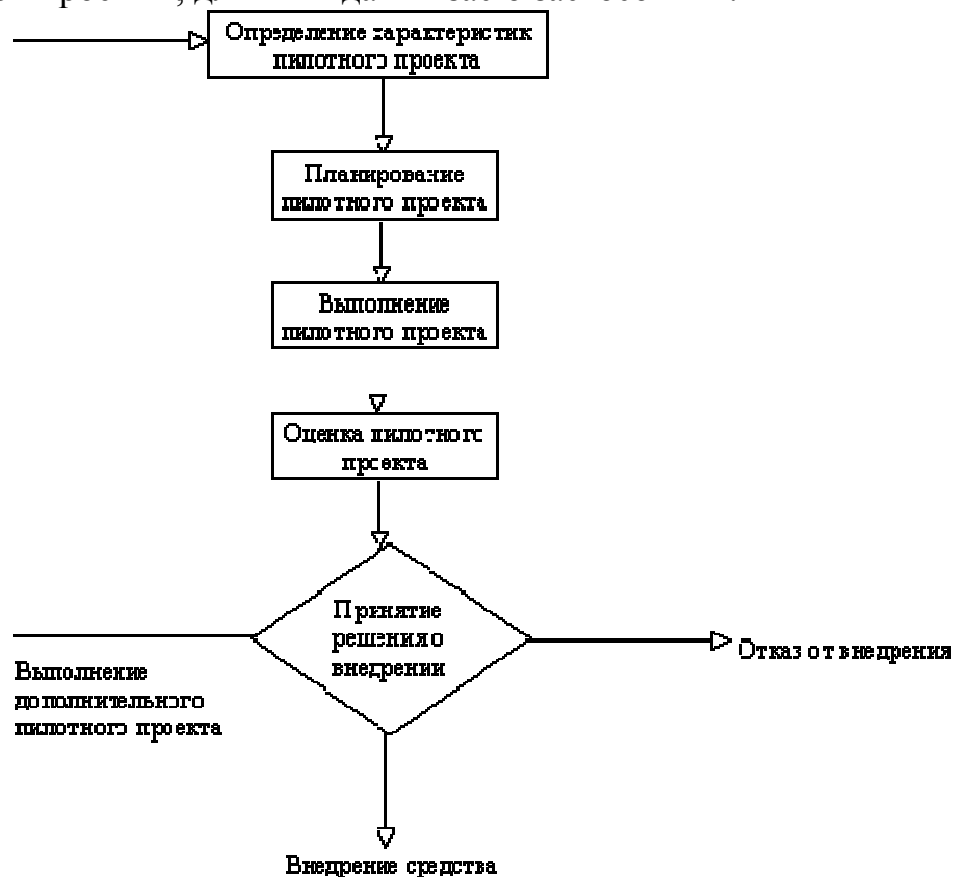
Первинне використання нової CASE-технології в пілотному проекті повинне ретельно плануватися і контролюватися. Пілотний проект включає наступні кроки (малюнок 4.4).

### Визначення характеристик пілотного проекту

Пілотний проект повинен володіти наступними характеристиками:

– Область застосування. Щоб полегшити остаточне визначення області застосування CASE-засобів, наочна область пілотного проекту повинна бути типовою для звичайної діяльності організації. Пілотний проект повинен допомогти визначити будь-яку додаткову технологію, навчання або підтримку, які необхідні для переходу від пілотного проекту до широкомасштабного використання засобу. В рамках цих обмежень пілотний проект повинен мати невеликий, але значущий розмір.

– Масштабованість. Результати, отримані в пілотному проекті, повинні показати масштабованість засобу. Мета - отримати чітке уявлення про масштаби проектів, для яких даний засіб застосовний.



Мал. 4.4. Кроки пілотного проекту

– Показність. Пілотний проект не повинен бути незвичайним або унікальним для організації. CASE-засобів повинен використовуватися для вирішення завдань, що відносяться до наочної області, що добре розуміється всією організацією.

– Критичність. Пілотний проект повинен мати істотну значущість, щоб опинитися в центрі уваги, але не повинен бути критичним для успішної діяльності організації в цілому. Необхідно усвідомлювати, що первинне впровадження нової технології має на увазі певний ризик. При виборі

пілотного проекту доводиться вирішувати наступну дилему: успіх незначного проекту може залишитися непоміченим, з іншого боку, провал значущого проекту може викликати надмірну критику.

- Авторитетність. Група фахівців, що беруть участь в проекті, повинна володіти високим авторитетом, при цьому результати проекту будуть серйозно сприйняті рештою співробітників організації.

- Характеристики проектної групи. Проектна група повинна володіти готовністю до нововведень, технічною зрілістю і прийнятним рівнем досвіду і знань в даній технології і наочній області. З іншого боку, група повинна відображати в мініатюрі характеристики всієї організації в цілому.

В більшості випадків існує баланс між бажанням реалізувати ідеальний пілотний проект і реальними обмеженнями організації. Організація повинна вибрати пілотний проект так, щоб, по-перше, спосіб використання CASE-засобів в нім співпадав з подальшими планами, і, по-друге, перераховані вище характеристики були збалансовані з реальними умовами організації.

Крім того, організація повинна враховувати тривалість пілотного проекту (і в цілому процесу впровадження). Дуже тривалий проект пов'язаний з ризиком втрати інтересу до нього з боку керівництва.

#### **Планування пілотного проекту**

Планування пілотного проекту повинне по можливості вписуватися в звичайний процес планування проектів в організації. План повинен містити наступну інформацію:

- цілі, завдання і критерії оцінки;
- персонал;
- процедури і угоди;
- навчання;
- графік і ресурси.

#### *Цілі, завдання і критерії оцінки*

Очікувані результати пілотного проекту повинні бути чітко визначені. Ступінь відповідності цим результатам є основою для подальшої оцінки проекту. Для визначення цілей, завдань і критеріїв оцінки необхідно виконати наступні дії:

- описати проект в термінах очікуваних результатів (тобто кінцевого продукту). Опис повинен включати форму уявлення і зміст результатів. Повинні бути чітко визначені договірні вимоги і відповідні стандарти.

- визначити загальні цілі проекту. Прикладом мети може бути визначення ступеня поліпшення якості проектної документації в результаті застосування CASE-засобів.

- визначити конкретні завдання, що реалізують поставлені цілі. Кожній меті можна поставити у відповідність одну або декілька конкретних завдань з кількісно оцінюваними результатами. Прикладом такого завдання може бути порівняльний аналіз якості документації, отриманої за допомогою CASE-засобів і без нього. Документація може включати специфікацію вимог ПЗ, високорівневі і детальні проектні специфікації.

– визначити критерії оцінки результатів. Щоб визначити ступінь успіху пілотного проекту, необхідно використовувати набір критеріїв, заснованих на згаданих вище завданнях. Прикладом критерію може бути ступінь несуперечності проектної документації і контрольованості виконання вимог до ПЗ. Значення критеріїв повинні порівнюватися з базовими значеннями, отриманими до виконання пілотного проекту.

#### *Персонал*

Фахівці, вибрані для участі в пілотному проекті, повинні мати відповідний авторитет і вплив і бути прихильниками нової технології. Група повинна включати як технічних фахівців, так і менеджерів, що зацікавлених в новій технології і розбираються в її використанні. Група повинна володіти високими здібностями до комунікації, знанням особливостей організаційних процесів і процедур, а також наочній області. Група не винна, проте, складатися повністю з фахівців вищої ланки, вона повинна представляти середній рівень організації.

Багато CASE-засобів забезпечують можливості, пов'язані з генерацією проектної документації і конфігураційним управлінням. Фахівці, пов'язані з цими і іншими суміжними аспектами розробки і супроводу ПЗ, також повинні бути включені до складу групи.

Після завершення пілотного проекту група повинна бути відкрита для обміну інформацією з рештою фахівців організації щодо можливостей нового засобу і досвіду, отриманого при його використанні. Може виявитися бажаним розосередити членів проектної групи по всій організації з метою розповсюдження їх досвіду і знань.

#### *Процедури і угоди*

Необхідно чітко визначити процедури і угоди, регулюючі використання CASE-засобів в пілотному проекті. Це завдання швидше за все може виявитися довшим і складнішим, чим очікується, при цьому може виявитися необхідним залучення сторонніх експертів. Прикладами процедур і угод, які можуть вплинути на успіх пілотного проекту, є методологія, технічні угоди (зокрема, по найменуваннях і структурі каталогів, стандарти проектування і програмування - див. підрозділ 1.3) і організаційні угоди (зокрема, облік використання ресурсів, авторизація, контроль змін, процедури експертизи і підготовки звітів, стандарти перевірки якості).

У пілотному проекті по можливості повинні використовуватися прийняті в організації процедури і угоди. З іншого боку, протягом пілотного проекту процедури і угоди мають тенденцію до розвитку і вдосконалення у міру накопичення досвіду застосування засобу. Існуючі процедури і угоди можуть виявитися неефективними або такими, що занадто обмежують. При цьому ті зміни, які пропонується в них вносити, повинні документуватися.

#### *Навчання*

Повинні бути визначені види і об'єм навчання, необхідного для виконання пілотного проекту. При плануванні навчання потрібно мати на увазі три види потреб: технічні, управлінські і мотиваційні. Ресурси, потрібні для навчання (учбові аудиторії і устаткування, викладачі і учбові матеріали),

повинні відповідати плану пілотного проекту.

Графік навчання повинен визначати як фахівців, що підлягають навчанню, так і види навчання, яке вони повинні пройти. Навчання, яке проводиться в період виконання проекту, повинне починатися щонайшвидше після початку проекту. Навчання засобам, процесам або методам, які не використовуватимуться протягом декількох місяців після початку проекту, повинне плануватися на той час, коли в них виникне реальна потреба.

Постачальники CASE-засобів зазвичай пропонують навчання використанню засобів, що поставляються ними. Крім цього, для деяких засобів може бути необхідне навчання методології. Деякі види навчання повинні виконуватися власними силами. Такі види навчання включають використання CASE-засобів в контексті процесів, що відбуваються в організації, а також в сукупності з іншими засобами в даному середовищі. Частина плану пілотного проекту, пов'язана з навчанням, повинна використовуватися як вхід для плану практичного впровадження.

При виборі необхідного навчання повинні прийматися до уваги наступні чинники:

- кваліфікація викладачів;
- відповідність навчання характеристикам конкретних груп фахівців (наприклад, оглядові курси для менеджерів, докладні курси для розробників);
- можливість проведення курсів безпосередньо на робочих місцях;
- можливість проведення розширених курсів;
- можливість підготовки власних викладачів.

#### *Графік і ресурси*

Повинен бути розроблений графік, що включає ресурси і терміни (етапи) проведення робіт. Ресурси включають персонал, технічні засоби, ПЗ і фінансування. Дані про персонал можуть визначати конкретних фахівців або вимоги до кваліфікації, необхідної для успішного виконання пілотного проекту. Фінансування повинне визначатися окремо по кожному виду робіт: придбання CASE-засобів, установка, навчання, окремі етапи проектування.

#### **Виконання пілотного проекту**

Пілотний проект повинен виконуватися відповідно до плану. Організаційна діяльність, пов'язана з виконанням пілотного проекту і підготовкою звітів, повинна виконуватися в установленому порядку. Пілотна природа проекту вимагає спеціальної уваги до питань придбання, підтримки, експертизи і оновлення версій. Ці питання розглядаються нижчим.

#### *Придбання, установка і інтеграція*

Після того, як CASE-засобів вибраний, воно повинне бути придбане, інтегроване в проектне середовище і настроєне відповідно до вимог пілотного проекту. Межі цієї діяльності залежать від тих дій, які мали місце в процесі оцінки і вибору, а також від ступеня модифікації засобу, необхідної для його використання в проекті.

Процес придбання може включати підготовку контракту, переговори, ліцензування і іншу діяльність, яка виходить за рамки даних рекомендацій.

Ця діяльність вимагає витрат часу і людських ресурсів, які повинні бути враховані при плануванні. План повинен передбачати можливість відмови від вибраного засобу на даному етапі із-за договірних розбіжностей.

Після того, як процес придбання завершений, засіб повинен бути встановлене, відтестувало і прийняте в експлуатацію. Тестування дозволяє переконатися, що поставлений продукт відповідає вимогам контракту, володіє необхідною повнотою і коректністю. Етап приймання може бути передбачений контрактом, його реальний термін може відрізнятись від того, який був передбачений спочатку в плані пілотного проекту. Особливу увагу необхідно приділити дотриманню всіх вимог постачальника до параметрів середовища функціонування CASE-засобів.

Після завершення приймання може потрібно настройка і інтеграція. Налаштування може включати модифікацію інтерфейсів, пов'язану з вимогами фахівців проектної групи, а також установкою прав доступу і привілеїв. Налаштування повинна залишатися в рамках тих можливостей, які надає само засіб. Не слід займатися модифікацією готових продуктів на рівні початкових код.

Якщо новий засіб повинен використовуватися в сукупності з деякими іншими засобами, необхідно визначити взаємодію засобів і необхідну інтеграцію. Для інтеграції нових засобів з тими, що існують може потрібно побудова спеціальних оболонок. Складна інтеграція може зажадати залучення сторонніх експертів.

#### *Підтримка*

Доступна підтримка повинна включати (за угодою) "гарячу лінію" постачальника і підтримку місцевого постачальника, підтримку в самій організації, контакти з досвідченими користувачами в інших організаціях і участь в роботі груп користувачів.

Внутрішня підтримка повинна здійснюватися фахівцями, знайомими з установкою засобів і роботою з ними. Існує декілька можливих варіантів отримання такої підтримки (наприклад, від фахівця даної організації, що має досвід попередньої роботи із засобом; учасників процесу оцінки і вибору або досвідченого консультанта). Такий тип підтримки повинен спеціальним чином плануватися і адмініструватися. Особлива увага повинна бути приділена засобам, що працюють в мережах або що володіють репозиторіями, що підтримують многопользовательську роботу.

#### *Періодичні експертизи*

Звичайні процедури експертизи проектів, що існують в організації, повинні виконуватися і для пілотного проекту, при цьому особлива увага повинна приділятися саме пілотним аспектам проекту. Крім цього, результати експертиз повинні служити мірою успішного використання CASE-засобів.

#### *Оновлення версій*

Користувачі CASE-засобів можуть чекати періодичного оновлення версій з боку постачальника протягом виконання пілотного проекту. При цьому необхідне ретельне відношення до інтеграції цих версій. Слід

заздалегідь оцінити вплив цих оновлень на хід проекту. Нові версії можуть як забезпечити нові можливості, так і породити нові проблеми. Наприклад, нова версія може зажадати видозміненого або додаткового навчання, а також може надати негативну дію на вже виконану до цього моменту роботу.

### **Оцінка пілотного проекту**

Після завершення пілотного проекту його результати необхідно оцінити і зіставити їх з початковими потребами організації, критеріями успішного впровадження CASE-засобів, базовими метриками і критеріями успіху пілотного проекту. Така оцінка повинна встановити можливі проблеми і найважливіші характеристики пілотного проекту, які можуть вплинути на придатність CASE-засобів для організації. Вона повинна також вказати проекти або структурні підрозділи усередині організації, для яких даний засіб є відповідним. Крім цього, оцінка може дати інформацію щодо вдосконалення процесу впровадження надалі.

В процесі оцінки пілотного проекту організація повинна визначити свою позицію по наступних трьох питаннях:

- Чи доцільно упроваджувати CASE-засобів ?
- Які конкретні особливості пілотного проекту привели до його успіху (або невдачі) ?
- Які проекти або підрозділи в організації могли б отримати вигоду від використання засобів ?

### **Ухвалення рішення про доцільність впровадження CASE-засобів**

На даному етапі процесу впровадження організація повинна зробити істотні інвестиції в CASE-засобів. Якщо засоби задовольнили або навіть перевищили очікування організації, то рішення про впровадження може бути ухвалене досить просто і швидко.

З іншого боку, може опинитися, що в рамках пілотного проекту засобу не виправдали тих очікувань, які на них поклалися, або ж в пілотному проекті вони використовувалися задовільно, проте досвід показав, що подальші вкладення в засоби не гарантують успіху.

Можливі чотири категорії результатів і відповідних дій:

– Пілотний проект потерпів невдачу, і його аналіз показав неадекватність очікувань організації. В цьому випадку організація може переглянути результати проекту в контексті реалістичніших очікувань.

– Пілотний проект потерпів невдачу, і його аналіз показав, що вибрані засоби не задовольняють потреби організації. В цьому випадку організація може ухвалити рішення не упроваджувати дані засоби, проте при цьому також переглянути свої потреби і підхід до оцінки і вибору CASE-засобів.

– Пілотний проект потерпів невдачу, і його аналіз показав наявність таких проблем, як невдалий вибір пілотного проекту, неадекватне навчання і недовлік ресурсів. В цьому випадку може опинитися достатньо складно ухвалити рішення про те, чи слід знов виконати пілотний проект, продовжити роботу по впровадженню або відмовитися від CASE-засобів. Проте, незалежно від ухваленого рішення, процес впровадження потребує

перегляду і підвищеної уваги.

– Пілотний проект завершився успішно, і визнано доцільним упроваджувати CASE-засоби в деяких підрозділах або, можливо, у всій організації в цілому. В цьому випадку наступним кроком є визначення найбільш відповідного масштабу впровадження.

У ряді випадків аналіз пілотного проекту може показати, що причиною невдачі з'явився більш ніж один чинник. Подальші спроби впровадження CASE-технології повинні чітко виявити всі причини невдачі. У екстремальних випадках ретельний аналіз може показати, що зараз організація просто не готова до успішного впровадження складних CASE-засобів. У такій ситуації організація може спробувати вирішити свої проблеми іншими засобами.

### **Особливості пілотного проекту**

Дуже важливо провести аналіз пілотного проекту з тим, щоб визначити його елементи, що є критичними для успіху, і визначити ступінь віддзеркалення цими елементами діяльності організації в цілому. Наприклад, якщо в пілотному проекті беруть участь самі кращі програмісти організації, він може закінчитися успішно навіть всупереч використанню CASE-засобів, а не завдяки ним. З іншого боку, CASE-засоби можуть бути застосовані для розробки застосування, для якого вони явно не підходять по своїх характеристиках. Проте, таке використання могло б вказати на область найбільш раціонального застосування засобів в даній організації.

Найважливіші характеристики пілотного проекту, що немає представницькими для організації в цілому, можуть включати наступні:

– Процеси в пілотному проекті в чому-небудь відрізняються від процесів у всій організації.

– Кваліфікація групи пілотного проекту не відображає кваліфікацію решти фахівців організації.

– Ресурси, виділені на виконання проекту, відрізняються від тих, які виділяються для звичайних проектів.

– Предметна область або масштаб проекту не відповідають іншим проектам.

### **Вигода від використання CASE-засобів**

Пілотний проект слід зіставити з діяльністю організації в цілому з тим, щоб визначити найбільш істотну схожість і відмінність. Наприклад, якщо найбільш зацікавлені і кваліфіковані учасники проекту зіткнулися з серйозними труднощами в освоєнні засобів, то менш зацікавленим і кваліфікованим програмістам з інших підрозділів буде потрібно істотно більше навчання.

Пілотний проект може також показати, що засоби доцільно використовувати для деяких класів проектів і недоцільно для інших. Наприклад, засіб формальної верифікації може підходити для життєво важливих застосувань і не підходити для менш критичних застосувань.

Перед розробкою плану переходу організація повинна оцінити



очікуваний ефект для різних підрозділів або класів проектів. При цьому слід враховувати, що деякі підрозділи можуть не володіти необхідною кваліфікацією або ресурсами для використання CASE-засобів.

#### **Ухвалення рішення про впровадження**

Можливим рішенням повинно бути одне з наступних:

- Упровадити засіб. Масштаб впровадження, що в цьому випадку рекомендується, повинен бути визначений в термінах структурних підрозділів і наочної області.

- Виконати додатковий пілотний проект. Такий варіант повинен розглядатися тільки в тому випадку, якщо залишилися конкретні нездолені питання щодо впровадження CASE-засобів в організації. Новий пілотний проект повинен бути таким, щоб відповісти на ці питання.

- Відмовитися від засобу. В цьому випадку причини відмови від конкретного засобу повинні бути визначені в термінах потреб організації або критеріїв, які залишилися незадоволеними. Перед тим, як продовжити діяльність по впровадженню CASE-засобів, потреби організації повинні бути переглянуті на предмет своєї обґрунтованості.

- Відмовитися від використання CASE-засобів взагалі. Пілотний проект може показати, що організація або не готова до впровадження CASE-засобів, або автоматизація даного аспекту процесу створення і супроводу ПЗ не дає ніякого ефекту для організації. В цьому випадку причини відмови від CASE-засобів повинні бути також визначені в термінах потреб організації або критеріїв, які залишилися незадоволеними. При цьому необхідно розуміти відмінність цього варіанту від попереднього, пов'язаного з недоліками конкретного засобу.

Результатом даного етапу є документ, в якому обговорюються результати пілотного проекту і деталізують рішення по впровадженню.

#### **4.4. Перехід до практичного використання CASE-засобів**

Процес переходу до практичного використання CASE-засобів починається з розробки і подальшої реалізації плану переходу. Цей план може відображати поетапний підхід до переходу, починаючи з ретельно вибраного пілотного проекту до проектів з істотно збільшеною різноманітністю характеристик.

##### **Розробка плану переходу**

План переходу повинен включати наступне:

- Інформацію щодо цілей, критеріїв оцінки, графіка і можливих ризиків, пов'язаних з реалізацією плану.

- Інформацію щодо придбання, установки і настройки CASE-засобів.

- Інформацію щодо інтеграції кожного засобу з існуючими засобами, включаючи як інтеграцію CASE-засобів один з одним, так і їх інтеграцію в процеси розробки і експлуатації ПЗ, що існують в організації.

- Очікувані потреби в навчанні і ресурси, використовувані в течію і після завершення процесу переходу.

- Визначення стандартних процедур використання засобів.

*Цілі, критерії оцінки, графік і ризики, пов'язані з планом переходу*

Дана інформація повинна включати наступне:

- Типи проектів, в яких кінець кінцем використовуватиметься засіб.
- Графік переходу до практичного використання засобу в окремих проектах, який включає необхідну підготовку до його широкого використання.
- Графік впровадження засобу в термінах кількості користувачів, включаючи необхідне навчання.
- Можливі ризики і непередбачені обставини.
- Джерела існуючих (базових) даних і метрики для оцінки змін, викликаних використанням засобів.

На додаток до сказаного, слід приділити особливу увагу питанням контролю змін. Ролі вищого керівництва, суб'єктів і об'єктів змін повинні бути уточнені в порівнянні з пілотним проектом, оскільки технологія підлягає широкому розповсюдженню в організації.

Мається на увазі, що план переходу успішно виконаний, коли не потрібно більше спеціального планування підтримки використання засобу. У цей момент використання засобу узгоджується з тим, що від нього очікувалося, і план роботи з ним включається в загальний план поточної підтримки ПЗ, що існує в організації.

*Придбання, установка і настройка засобів*

Придбання, установка і настройка, виконані в рамках пілотного проекту, можуть потрібно в ширшому масштабі. При цьому необхідна наступна інформація:

- Сукупність програмних компонент, документація і навчання, яких необхідно набувати для кожної окремої платформи.
- Механізм отримання нових версій.
- Настройка засобу, необхідна для виконання процедур, що існують в організації, і угод.
- Обличчя або підрозділ, відповідальний за установку, інтеграцію, настройку і експлуатацію засобу.
- План конвертації даних і зняття старих засобів з експлуатації.

Завдання придбання, установки і настройки повинні бути щонайшвидше передані з групи пілотного проекту в існуючу службу системної підтримки ПЗ організації.

*Інтеграція засобу з існуючими засобами і процесами*

Інтеграція нового засобу з існуючими засобами і процесами є важливим кроком в повномасштабному впровадженні засобу. В більшості випадків така інтеграція в процесі пілотного проектування не здійснюється, проте накопичувана в цьому процесі інформація може допомогти в розробці планів інтеграції. Для планування інтеграції необхідна наступна інформація:

- Найменування і версії існуючих засобів, з якими повинен інтегруватися новий засіб.
- Описи даних, які повинні спільно використовуватися новим і

існуючими засобами, а також попередня інформація про джерела цих даних.

- Описи інших взаємозв'язків між новим і існуючими засобами (таких, як зв'язки по передачі управління і порядку використання), а також попередня інформація про механізми підтримки цих взаємозв'язків.

- Оцінки витрат, термінів і ризик, пов'язаних з інтеграцією (і, можливо, переходом від існуючих засобів і даних).

- Визначення способу впровадження даного засобу в діяльність по вдосконаленню існуючих процесів.

- Очікувані зміни в існуючих процесах і продуктах, використання нового засобу, що є слідством, і оцінювані, по можливості, кількісно.

Ризик, пов'язаний з інтеграцією нового засобу з існуючими засобами і процесами, знижується, якщо потреби в інтеграції враховуються в процесі оцінки і вибирання засобу.

*Навчання і ресурси, використовувані в течію і після завершення процесу переходу*

Дана інформація повинна включати наступне:

- Персонал (включаючи користувачів, адміністраторів і інтеграторів), що потребує навчання використанню засобу.

- Вид навчання, необхідного для кожної категорії користувачів і обслуговуючого персоналу, з урахуванням особливої важливості навчання сумісному використанню різних засобів, а також методам і процесам, пов'язаним з даними засобами.

- Вид навчання, необхідного для різних фахівців (наприклад, для групи тестування і незалежної служби сертифікації).

- Частота навчання.

- Види і доступність підтримки.

*Визначення стандартів і процедур використання засобів*

План переходу повинен визначати наступні стандарти і процедури використання засобів:

- Керівництво по моделюванню і проектуванню.

- Угоди по привласненню імен.

- Процедури контролю якості і процесів приймання, включаючи розклад експертиз і використовувані методології.

- Процедури резервного копіювання, захисту мастер-копій і конфігурації бази даних.

- Процедури інтеграції з існуючими засобами і базами даних.

- Процедури сумісного використання даних і контролю цілісності БД.

- Стандарти і процедури забезпечення секретності.

- Стандарти документування.

Стандарти використання CASE-засобів, вироблені під час пілотного проекту, повинні використовуватися як відправна крапка для розробки повнішого набору стандартів використання засобів в даній організації (див. підрозділ 1.3). При цьому повинен враховуватися досвід учасників пілотного проекту.

## **Реалізація плану переходу**

Реалізація плану переходу вимагає постійного моніторингу використання CASE-засобів, забезпечення поточної підтримки, супроводу і оновлення засобів в міру необхідності.

### *Періодичні експертизи*

Досягнуті результати повинні періодично піддаватися експертизі відповідно до графіка, план переходу повинен коректуватися при необхідності. Постійна увага повинна приділятися ступеню задоволення потреб організації і критеріїв успішного впровадження CASE-засобів.

Періодичні експертизи повинні продовжуватися і після завершення процесу впровадження. Такі експертизи можуть аналізувати метрики і іншу інформацію, що отримується в процесі роботи з CASE-засобами, щоб визначати, наскільки добре вони продовжують виконувати необхідні функції. Такі експертизи можуть також вказати на необхідність додаткової модифікації процесів.

### *Поточна підтримка*

Поточна підтримка необхідна для наступного:

- Відповідей на питання, зв'язані з використанням засобів.
- Передачі інформації про досягнуті успіхи і отримані уроки іншим фахівцям організації.
- Модифікації і вдосконалення стандартів, угод і процедур, зв'язаних з використанням засобу.
- Інтеграції нових засобів з тими, що існують і супровід інтегрованих засобів у міру появи нових версій.
- Допомозі новим співробітникам в освоєнні засобів і пов'язаних з ними процедур.
- Планування і контролю оновлення версій.
- Планування впровадження нових можливостей засобів в організаційні процеси.

### **Дії, що виконуються в процесі переходу**

Для підтримки процесу переходу до практичного використання засобів бажане виконання наступних дій:

- Підтримка поточного навчання. Потреба в навчанні може виникати періодично унаслідок появи нових версій засобів або залучення до проекту нових співробітників.
- Підтримка ролевих функцій, пов'язаних з процесом впровадження. Оскільки впровадження CASE-засобів приводить до змін в культурі організації, необхідно в процесі впровадження виділити ряд ключових ролей (такі, як керівництво вищого рівня, проектна група і цільові групи).
- Методики управління оновленням версій. Ці методики можуть бути пов'язані з оновленням версій, процедурами установки, процедурами контролю якості для оцінки нових версій, процедурами оновлення бази даних, конфігурацією версій і середовищем підтримки (інші засоби, операційна система і так далі).

– Вільний доступ до інформації. Повинні бути визначені механізми, що забезпечують вільний доступ до інформації про досвід впровадження і уроки, що витягують з цього, включаючи дошки оголошень, інформаційні бюлетені, призначені для користувача групи, семінари і публікації.

– Налагодження тісної робочої взаємодії з постачальником. Така взаємодія дозволяє організації бути в курсі планів постачальника і забезпечувати оперативне задоволення своїх вимог.

Для успішного впровадження CASE-засобів в організації істотно важливою є послідовність в їх застосуванні. Оскільки більшість систем розробляються колективно, необхідно визначити характер майбутнього використання засобів як окремими розробниками, так і групами. Використання стандартних процедур дозволить забезпечити плавний перехід між окремими стадіями ЖЦ ПЗ.

Як правило, всі розуміють, що навчання є центральною ланкою, що забезпечує нормальне використання CASE-засобів в організації. Проте, досить поширена помилка полягає в тому, що проводиться початкове навчання для групи непідготовлених користувачів, а потім все обмежується мінімальним поточним навчанням. Учасники пілотного проекту, що отримали початкове навчання, можуть бути високо кваліфікованими ентузіастами нової технології, прагнучими використовувати її в що б те не стало. З іншого боку, для розробників, які братимуть участь в проекті надалі, може потрібно інтенсивніше і глибше навчання, а також поточна підтримка у використанні засобу.

На додаток до цього слід зазначити, що кожна категорія персоналу (наприклад, адміністратори засобів, служба підтримки робочих місць, інтегратори засобів, служба супроводу і розробники застосувань) потребує різного навчання.

Навчання не повинне замикатися тільки на користувачах CASE-засобів, навчатися повинні також ті співробітники організації, на діяльність яких так чи інакше робить вплив використання CASE-засобів.

В процесі подальшого використання засобів в організації навчання повинне стати частиною процесу орієнтації при наймі нових співробітників і залученні співробітників до проектів, в яких використовуються CASE-засобів. Навчання повинне стати невід'ємною складовою частиною нормативних матеріалів, що стосуються діяльності організації, які пропонуються новим співробітникам.

Одна загальна помилка, яка робиться в процесі переходу, полягає в недооцінці ресурсів, необхідних для підтримки постійного використання складних CASE-засобів. Зростання необхідних ресурсів викликається трьома причинами:

- Складністю засобів
- Частотою появи нових версій
- Взаємодією між засобами і зовнішнім середовищем.

Складність засобів приводить до зростання потреб в ретельному і

продуманому навчанні. Крім того, багато CASE-засобів можуть використовуватися тільки кваліфікованими фахівцями, що уміють супроводжувати проектні бази даних і оперативно реагувати на виникаючі проблеми. Висока частота оновлення версій засобів може привести до виникнення нетривіальних проблем, які часто випускаються з уваги. Такі оновлення зазвичай згубно відбиваються на жорстких планах і графіках роботи. Взаємодія між засобами і зовнішньою по відношенню до них середовищем також може іноді породжувати деякі проблеми. Мається на увазі той факт, що хоча багато засобів досягли рівня мінімальної несумісності даних між окремими версіями, проблеми забезпечення сумісності з іншими елементами зовнішнього середовища залишаються

### **Оцінка результатів переходу**

Програма постійної оцінки якості і продуктивності ПЗ має важливе значення для наступного:

- Визначення ступеня вдосконалення процесів
- Попередження можливих стратегічних прорахунків
- Своєчасної відмови від використання застарілої технології.

Щоб визначити, наскільки ефективний новий CASE-засобів підвищує продуктивність і/або якість, організація повинна спиратися на деякі базові дані. На жаль, лише небагато організацій в даний час накопичують дані для реалізації програми поточної кількісної оцінки і удосконалення процесів. Для доказу ефективності CASE-засобів і їх можливостей покращувати продуктивність необхідні такі базові метричні дані, як:

- Використаний час
- Час, виділений персонально для конкретних фахівців
- Розмір, складність і якість ПЗ
- Зручність супроводу.

Метрична оцінка повинна починатися з реальної оцінки поточного стану середовища ще до початку впровадження CASE-засобів і підтримувати процедури постійного накопичення даних.

Період часу, протягом якого виконується кількісна оцінка дії, що надається впровадженням CASE-засобів, є вельми значущим величиною з погляду визначення ступеня успішності переходу. Деякі організації, CASE-засобів, що успішно упровадили кінець кінцем, зіткнулися з короткочасними негативними ефектами на початку процесу. Інші, успішно почавши, недооцінили довготривалі витрати на супровід і навчання. Внаслідок цього, найбільш прийнятний часовий інтервал для оцінки ступеня успішності впровадження повинен бути достатньо великим, щоб подолати будь-які негативні ефекти на початковому етапі, а також змодельовати майбутні довготривалі витрати. З іншого боку, даний інтервал повинен відповідати цілям організації і очікуваним результатам.

Кінець кінцем, досвід, отриманий при впровадженні CASE-засобів, може частково змінити цілі організації і очікування, що покладаються на CASE-засобів. Наприклад, організація може зробити вивід, що засоби

доцільно використовувати для більшого або меншого круга користувачів і процесів в циклі створення і супроводу ПЗ. Такі зміни в очікуваннях часто можуть дати позитивні результати, але можуть також привести до внесення відповідних коректив до визначення ступеня успішного впровадження CASE-засобів в даній організації.

Результатом даного етапу є впровадження CASE-засобів в повсякденну практику організації, при цьому більше не вимагається якого-небудь спеціального планування. Крім того, підтримка CASE-засобів включається в план поточної підтримки ПЗ в даній організації.

## 5. Характеристики CASE-засобів

### 5.1. Silverrun+JAM

#### 5.1.1. Silverrun

CASE-засобом Silverrun американської фірми Computer Systems Advisers, Inc. (CSA) використовується для аналізу і проектування ІС бизнес-класа [22] і орієнтовано більшою мірою на спіральну модель ЖЦ. Воно застосовне для підтримки будь-якої методології, заснованої на роздільній побудові функціональної і інформаційної моделей (діаграм потоків даних і діаграм "суть-зв'язок").

Настроювання на конкретну методологію забезпечується вибором необхідної графічної нотації моделей і набору правил перевірки проектних специфікацій. У системі є готові настройки для найбільш поширених методологій: DATARUN (основна методологія, підтримувана Silverrun), Gane/Sarson, Yourdon/DeMarco, Merise, Ward/Mellor, Information Engineering. Для кожного поняття, введеного в проєкті є можливість додавання власних описувачів. Архітектура Silverrun дозволяє нарощувати середовище розробки в міру необхідності.

#### *Структура і функції*

Silverrun має модульну структуру і складається з чотирьох модулів, кожен з яких є самостійним продуктом і може отримуватися і використовуватися без зв'язку з рештою модулів.

Модуль побудови моделей бизнес-процесов у формі діаграм потоків даних (BPM - Business Process Modeler) дозволяє моделювати функціонування обстежуваної організації або створюваною ІС. У модулі BPM забезпечена можливість роботи з моделями великої складності: автоматична перенумерація, робота з деревом процесів (включаючи візуальне перетягання гілок), від'єднання і приєднання частин моделі для колективної розробки. Діаграми можуть зображатися в декількох зумовлених нотаціях, включаючи Yourdon/DeMarco і Gane/Sarson. Є також можливість створювати власні нотації, зокрема додавати в число дескрипторів, що зображаються на схемі, визначені користувачем поля.

Модуль концептуального моделювання даних (ERX - Entity-Relationship eXpert) забезпечує побудова моделей даних "суть-зв'язок", не прив'язаних до конкретної реалізації. Цей модуль має вбудовану експертну систему, що дозволяє створити коректну нормалізовану модель даних за допомогою відповідей на змістовні питання про взаємозв'язок даних. Можлива автоматична побудова моделі даних з описів структур даних. Аналіз функціональних залежностей атрибутів дає можливість перевірити відповідність моделі вимогам третьої нормальної форми і забезпечити їх виконання. Перевірена модель передається в модуль RDM.

Модуль реляційного моделювання (RDM - Relational Data Modeler) дозволяє створювати деталізовані моделі "суть-зв'язок", призначені для реалізації в реляційній базі даних. У цьому модулі документуються всі конструкції, пов'язані з побудовою бази даних: індекси, тригери, процедури, що зберігаються, і так далі Гнучка змінна нотація і розширюваність



репозиторія дозволяють працювати за будь-якою методологією. Можливість створювати підсхеми відповідає підходу ANSI SPARC до представлення схеми бази даних. На мові підсхем моделюються як вузли розподіленої обробки, так і призначені для користувача уявлення. Цей модуль забезпечує проектування і повне документування реляційних баз даних.

Менеджер репозиторія робочої групи (WRM - Workgroup Repository Manager) застосовується як словник даних для зберігання загальної для всіх моделей інформації, а також забезпечує інтеграцію модулів Silverrun в єдине середовище проектування.

Платою за високу гнучкість і різноманітність образотворчих засобів побудови моделей є такий недолік Silverrun, як відсутність жорсткого взаємного контролю між компонентами різних моделей (наприклад, можливості автоматичного розповсюдження змін між DFD різних рівнів декомпозиції). Слідуює, проте, відзначити, що цей недолік може мати істотне значення тільки у разі використання каскадної моделі ЖЦ ПЗ.

#### *Взаємодія з іншими засобами*

Для автоматичної генерації схем баз даних у Silverrun існують мости до найбільш поширених СУБД: Oracle, Informix, DB2, Ingres, Progress, SQL Server, SQLBase, Sybase. Для передачі даних в засоби розробки застосувань є мости до мов 4GL: JAM, PowerBuilder, SQL Windows, Uniface, NewEra, Delphi. Всі мости дозволяють завантажити в Silverrun RDM інформацію з каталогів відповідних СУБД або мов 4GL. Це дозволяє документувати, перепроєктувати або переносити на нові платформи бази даних, що вже знаходяться в експлуатації, і прикладні системи. При використанні моста Silverrun розширює свій внутрішній репозиторій специфічними для цільової системи атрибутами. Після визначення значень цих атрибутів генератор застосувань переносить їх у внутрішній каталог середовища розробки або використовує при генерації коду на мові SQL. Таким чином можна повністю визначити ядро бази даних з використанням всіх можливостей конкретної СУБД: тригерів, процедур, що зберігаються, обмежень посилюючої цілісності. При створенні застосування на мові 4GL дані, перенесені з репозиторія Silverrun, використовуються або для автоматичної генерації інтерфейсних об'єктів, або для швидкого їх створення уручну.

Для обміну даними з іншими засобами автоматизації проектування, створення спеціалізованих процедур аналізу і перевірки проектних специфікацій, складання спеціалізованих звітів відповідно до різних стандартів в системі Silverrun є три способи видачі проектної інформації в зовнішні файли:

– Система звітів. Можна, визначивши вміст звіту по репозиторію, видати звіт в текстовий файл. Цей файл можна потім завантажити в текстовий редактор або включити в інший звіт;

– Система експорту/імпорту. Для повнішого контролю над структурою файлів в системі експорту/імпорту є можливість визначити не тільки вміст експортного файлу, але і роздільники записів, полів в записах, маркери почала і кінця текстових полів. Файли з вказаною структурою можна

не тільки формувати, але і завантажувати в репозиторій. Це дає можливість обмінюватися даними з різними системами: іншими CASE-засобами, СУБД, текстовими редакторами і електронними таблицями;

– Зберігання репозиторія в зовнішніх файлах через ODBC-драйвери. Для доступу до даним репозиторія з найбільш поширених систем управління базами даних забезпечена можливість зберігати всю проектну інформацію безпосередньо у форматі цих СУБД.

#### *Групова робота*

Групова робота підтримується в системі Silverrun двома способами:

– У стандартній однопользовательской версії є механізм контрольованого розділення і злиття моделей. Розділивши модель на частини, можна роздати їх декільком розробникам. Після детального доопрацювання моделі об'єднуються в єдині специфікації;

– Мережева версія Silverrun дозволяє здійснювати одночасну групову роботу з моделями, що зберігаються в мережевому репозиторії на базі СУБД Oracle, Sybase або Informix. При цьому декілька розробників можуть працювати з однією і тією ж моделлю, оскільки блокування об'єктів відбувається на рівні окремих елементів моделі.

#### *Середовище функціонування*

Є реалізації Silverrun трьох платформ - MS Windows, Macintosh і OS/2 Presentation Manager - з можливістю обміну проектними даними між ними.

Для функціонування в середовищі Windows необхідно мати комп'ютер з процесором моделі не нижче i486 і оперативну пам'ять об'ємом не менше 8 Мб (рекомендується 16 Мб). На диску повна інсталяція Silverrun займає 20 Мб.

### **5.1.2. JAM**

Засіб розробки застосувань JAM [28] (JYACC's Application Manager) - продукт фірми JYACC (США). В даний час поставляється версія JAM 7 і готується до виходу JAM 8.

Основною межею JAM є його відповідність методології RAD, оскільки він дозволяє достатньо швидко реалізувати цикл розробки застосування, що полягає у формуванні чергової версії прототипу застосування з урахуванням вимог, виявлених на попередньому кроці, і пред'явити його користувачеві.

#### *Структура і функції*

JAM має модульну структуру і складається з наступних компонент:

- Ядро системи;
- JAM/DBi - спеціалізовані модулі інтерфейсу до СУБД (JAM/DBi-Oracle, JAM/DBi-Informix, JAM/DBi-ODBC і так далі);
- JAM/RW - модуль генератора звітів;
- JAM/CASEi - спеціалізовані модулі інтерфейсу до CASE-засобів (JAM/CASE-TeamWork, JAM/CASE-Innovator і так далі);
- JAM/TPi - спеціалізовані модулі інтерфейсу до менеджерів транзакцій (наприклад, JAM/TPi-Server TUXEDO і так далі);
- Jterm - спеціалізований емулятор X-терміналу.

Ядро системи (власне, сам JAM) є закінченим продуктом і може самостійно використовуватися для розробки застосувань. Решта всіх модулів є додатковими і самостійно використовуватися не можуть.

Ядро системи включає наступні основні компоненти:

- редактор екранів. До складу редактора екранів входять: середовище розробки екранів, візуальний репозиторій об'єктів, власна СУБД JAM - JDB, менеджер транзакцій, відладчик, редактор стилів;
- редактор меню;
- набір допоміжних утиліт;
- засоби виготовлення промислової версії застосування.

При використанні JAM розробка зовнішнього інтерфейсу застосування є візуальним проектуванням і зводиться до створення екранних форм шляхом розміщення на них інтерфейсних конструкцій і визначення екранних полів введення/виводу інформації. Проектування інтерфейсу в JAM здійснюється за допомогою редактора екранів. Застосування, розроблені в JAM, мають багатовіконний інтерфейс. Розробка окремого екрану полягає в розміщенні на ній інтерфейсних елементів, можливого (але не обов'язковою) їх угрупованні і конкретизації різних їх властивостей, що включають візуальні характеристики (позиція, розмір, колір, шрифт і тому подібне), поведінкові характеристики (багатообразні фільтри, формати, захист від введення і тому подібне) і ряд властивостей, орієнтованих на роботу з БД.

*Редактор меню* дозволяє розробляти і відлажувати системи меню. Реалізована можливість побудови піктографічних меню (так звані toolbar). Призначення кожного конкретного меню тому або іншому об'єкту застосування здійснюється в редакторі екранів.

У ядро JAM вбудована однопользовательская реляційна СУБД JDB. Основним призначенням JDB є прототипування застосувань в тих випадках, коли робота з штатною СУБД неможлива або недоцільна. У JDB реалізований необхідний мінімум можливостей реляційних СУБД за винятком індексів, процедур, що зберігаються, тригерів і уявлень (view). За допомогою JDB можна побудувати БД, ідентичну цільовою БД (з точністю до відсутніх в JDB можливостей) і розробити значну частину застосування.

*Відладчик* дозволяє проводити комплексну відладку застосування, що розробляється. Здійснюється трасування всіх подій, що виникають в процесі виконання застосування.

Утиліти JAM включають три групи:

- конвертори файлів екранів JAM в текстових. JAM зберігає екрани у вигляді двійкових файлів власного формату. У ряді випадків (наприклад для виготовлення програмної документації проекту) необхідний текстовий опис екранів;
- конфігурація пристроїв введення/виводу. JAM і застосування, побудовані з його допомогою, не працюють безпосередньо з пристроями введення/виводу. Замість цього JAM звертається до логічних пристроїв введення/виводу (клавіатура, термінал, звіт). Відображення логічних

пристроїв у фізичних здійснюється за допомогою засобів конфігурації;

- обслуговування бібліотек екранів (традиційні операції з бібліотеками).

Одним з додаткових модулів JAM є генератор звітів. Компонівка звіту здійснюється в редакторі екранів JAM. Опис роботи звіту здійснюється за допомогою спеціальної мови. Генератор звітів дозволяє визначити дані, що виводяться в звіт, угруповання інформації, що виводиться, форматування виводу і ін.

Застосування, розроблені з використанням JAM, не вимагають так званих виконавських (run-time) систем і можуть бути виготовлені у вигляді виконуваних модулів. Для цього розробник повинен мати компілятор C і редактор зв'язків. Для виготовлення промислової версії до складу JAM входить файл збірки (makefile), початкові тексти (на мові C) ряду модулів застосування і необхідні бібліотеки.

JAM містить вбудовану мову програмування JPL (JAM Procedural Language), за допомогою якого у разі потреби можна написати модулі, що реалізують специфічні дії. Дана мова є такою, що інтерпретується, що спрощує відладку. Існує можливість обміну інформацією між середовищем візуально побудованого застосування і такими модулями. Крім того, в JAM реалізована можливість підключення зовнішніх модулів, написаних на якій-небудь мові, сумісним по викликах функцій з мовою C.

З погляду реалізації логіки застосування JAM є подієво-орієнтованою системою. У JAM визначений набір подій, що включає відкриття і закриття вікон, натиснення клавіші клавіатури, спрацьовування системного таймера, отримання і передача управління кожним елементом екрану. Розробник реалізує логіку застосування шляхом визначення обробника кожної події. Наприклад, обробник події "натиснення кнопки на екрані" (мишею або за допомогою клавіатури) може відкрити наступне екранне вікно. Обробниками подій в JAM можуть бути як вбудовані функції JAM, так і функції, написані розробником на C або JPL. Набір вбудованих функцій включає більше 200 функцій різного призначення. Вбудовані функції доступні для викликів з функцій, написаних як на JPL, так і на C.

Промислова версія застосування, розробленого за допомогою JAM, включає наступні компоненти:

- виконуваний модуль інтерпретатора застосування. У цей модуль можуть бути вбудовані функції, написані розробниками на мовах 3-го покоління;

- екрани, складові само застосування (можуть поставлятися у вигляді окремих файлів, у складі бібліотек екранів або ж бути вбудовані в тіло інтерпретатора);

- зовнішні JPL-модулі. Можуть поставлятися у вигляді текстових файлів або у вигляді, що прекомпілированном, причому зовнішні JPL-модулі, що прекомпілированные, можуть бути як у вигляді окремих файлів, так і у складі бібліотек екранів;

– файли конфігурації застосування - файли конфігурації клавіатури і терміналу, файл системних повідомлень, файл загальної конфігурації.

#### *Взаємодія з іншими засобами*

Безпосередню взаємодію з СУБД реалізують модулі JAM/DBi (Data Base interface). Способи реалізації взаємодії в JAM розділяються на два класи: ручні і автоматичні. При ручному способі розробник застосування самостійно пише запити на SQL, в яких як джерелами, так і адресатами прийому результатів виконання запиту можуть бути як інтерфейсні елементи візуально спроектованого зовнішнього рівня, так і внутрішні, невидимі для кінцевого користувача змінні. Автоматичний режим, що реалізовується менеджером транзакцій JAM, здійснює для типових і найбільш поширених видів операцій з БД, так званих QBE (Query By Example - запити за зразком), з урахуванням достатніх складних взаємозв'язків між таблицями БД і автоматичним управлінням атрибутами екранних полів введення/виводу залежно від вигляду транзакції (читання, запис і так далі), в якій бере участь запит, що згенерував.

JAM дозволяє будувати застосування для роботи більш ніж з 20 СУБД: ORACLE, Informix, Sybase, Ingres, InterBase, NetWare SQL Server, Rdb, DB2, ODBC-совместимые СУБД і ін.

Відмінною рисою JAM є високий рівень переносимості застосувань між різними платформами (MS DOS/MS Windows, SUNOS, Solaris (i80x86, SPARC), HP-UX, AIX, VMS/Open VMS і ін.). Може потрібно лише "перемальовувати" статичні текстові поля на екранах з російським текстом при перенесенні між середовищами DOS-Windows-UNIX. Крім того, переносимість полегшується тим, що в JAM застосування розробляються для віртуальних пристроїв введення/виводу, а не для фізичних. Таким чином при перенесенні застосування з платформи на платформу, як правило, потрібно лише визначити відповідність між фізичними пристроями введення/виводу і їх логічними уявленнями для застосування.

Використання SQL як засіб взаємодії з СУБД також створює передумови для забезпечення переносимості між СУБД. За умови перенесення структури самою БД у ряді випадків застосування можуть не вимагати ніякої модифікації, за винятком ініціалізації сеансу роботи. Така ситуація може скластися в тому випадку, якщо в застосуванні не використовувалися специфічні для тієї або іншої СУБД розширення SQL.

При зростанні навантаження на систему і складності вирішуваних завдань (распределенность і гетерогенність використовуваних ресурсів, кількість одночасно підключених користувачів, складність логіки застосування) застосовується триланкова модель архітектури "клієнт-сервер" з використанням менеджерів транзакцій. Компоненти JAM/TPi-Client і JAM/TPi-Server дозволяють досить просто перейти на триланкову модель. При цьому ключову роль грає модуль JAM/TPi-Server, оскільки основна трудність впровадження триланкової моделі полягає в реалізації логіки застосування в сервісах менеджерів транзакцій.

*Інтерфейс JAM/CASE* подібний до інтерфейсу до СУБД і дозволяє

здійснити обмін інформацією між репозиторієм об'єктів JAM і репозиторієм CASE-засобів аналогічно тому, як структура БД імпортується в репозиторій JAM безпосередньо з БД. Відмінність полягає в тому, що у разі інтерфейсу до CASE цей обмін є двонаправленим. Окрім модулів JAM/CASEi, існує також модуль JAM/CASEi Developer's Kit. За допомогою цього модуля можна самостійно розробити інтерфейс (тобто спеціалізований модуль JAM/CASEi) для конкретного CASE-засобів, якщо готового модуля JAM/CASEi для нього не існує.

Міст (інтерфейс) SILVERRUN-RDM <-> JAM реалізує взаємодія між CASE-засобівом Silverrun і JAM (перенесення схеми бази даних і екранних форм застосування між CASE-засобівом SILVERRUN-RDM і JAM версії 7.0). Даний програмний продукт має 2 режими роботи:

- прямий режим (Silverrun-RDM->JAM) призначений для створення об'єктів CASE-словаря і елементів репозиторія JAM на основі представлення схем в SILVERRUN-RDM. У цьому режимі міст дозволяє, виходячи з представлення моделей даних інтерфейсу в SILVERRUN-RDM, проводити генерацію екранів і елементів репозиторія JAM. Міст перетворить таблиці і відносини реляційних схем RDM в послідовність об'єктів JAM відповідних типів. Методика побудови моделей даних інтерфейсу в SILVERRUN-RDM припускає застосування механізму підсхем для прототипування екранів застосування. По опису кожній з підсхем RDM міст генерує екранну форму JAM;

- зворотний режим (JAM->Silverrun-RDM) призначений для перенесення модифікацій об'єктів CASE-словаря в реляційну модель SILVERRUN-RDM.

Режим реінжиниринга дозволяє переносити модифікації всіх властивостей екранів JAM, імпортованих раніше з RDM, в схему Silverrun. На цьому етапі для контролю цілісності бази даних не допускаються зміни схеми у вигляді додавання або видалення таблиць і полів таблиць.

#### *Групова робота*

Ядро JAM має вбудований інтерфейс до засобів конфігураційного управління (PVCS на платформі Windows і SCCS на платформі UNIX). Під управлінням цих систем передаються бібліотеки екранів і/або репозиторії. За відсутності таких систем JAM самостійно реалізує частину функцій підтримки групової розробки.

Використання PVCS (див. підрозділ [5.6](#)) є переважнішим в порівнянні з SCCS, оскільки дозволяє організувати єдиний архів модулів проекту для всіх платформ. Оскільки JAM на платформі UNIX не має прямого інтерфейсу до архівів PVCS, то вибірка модулів з архіву і повернення їх в архів проводяться з використанням PVCS Version Manager. На платформі MS-Windows JAM має вбудований інтерфейс до PVCS і дій з вибірки/повернення проводяться безпосередньо з середовища JAM.

#### *Середовище функціонування*

JAM, як середовище розробки, і застосування, побудовані з його використанням, не є ресурсоемними системами. Наприклад, на платформі

MS-Windows досить мати 8MB оперативної пам'яті і 50 MB дискового простору для середовища розробки. На UNIX-платформах вимоги до апаратури визначаються самою операційною системою.

## 5.2. Vantage Team Builder (Westmount I-CASE) + Uniface

### 5.2.1. Vantage Team Builder (Westmount I-CASE)

Vantage Team Builder [14] є інтегрованим програмним продуктом, орієнтованим на реалізацію каскадної моделі ЖЦ ПЗ і підтримку повного ЖЦ ПЗ.

#### *Структура і функції*

Vantage Team Builder забезпечує виконання наступних функцій:

- проектування діаграм потоків даних, "суть-зв'язок", структур даних, структурних схем програм і послідовностей екранних форм;
- проектування діаграм архітектури системи - SAD (проектування складу і зв'язку обчислювальних засобів, розподіли завдань системи між обчислювальними засобами, моделювання відносин типу "клієнт-сервер", аналіз використання менеджерів транзакцій і особливостей функціонування систем в реальному часі);
- генерація коду програм на мові 4GL цільової СУБД з повним забезпеченням програмного середовища і генерація SQL-кода для створення таблиць БД, індексів, обмежень цілісності і процедур, що зберігаються;
- програмування на мові C з вбудованим SQL;
- управління версіями і конфігурацією проекту;
- многопользовательский доступ до репозиторія проекту;
- генерація проектної документації по стандартних і індивідуальних шаблонах;
- експорт і імпорт даних проекту у форматі CDIF (CASE Data Interchange Format).

Vantage Team Builder поставляється в різних конфігураціях залежно від використовуваних СУБД (ORACLE, Informix, Sybase або Ingres) або засобів розробки застосувань (Uniface). Конфігурація Vantage Team Builder for Uniface відрізняється від останніх деякою мірою орієнтації на спіральну модель ЖЦ ПЗ за рахунок можливостей швидкого прототипування, Uniface, що надаються. Для опису проекту ІС використовується достатньо великий набір діаграм, конкретні варіанти якого для найбільш поширених конфігурацій приведені нижче в таблиці.

Тип діаграми	Позначення	Vantage Team Builder for ORACLE	Vantage Team Builder for Informix	Vantage Team Builder for Uniface
Суть-зв'язок	ERD	+	+	+
Потоків даних	DFD	+	+	+
Структур	DSD	+	+	+

даних				
Архітектура системи	SAD	+	+	+
Потоків управління	CSD	+	+	+
Типів даних	DTD	+	+	+
Структури меню	MSD	+		
Послідовності блоків	BSD	+		
Послідовності форм	FSD		+	+
Вмісту форм	FCD		+	+
Переходів станів	STD	+	+	+
Структурних схем	SCD	+	+	+

При побудові всіх типів діаграм забезпечується контроль відповідності моделей синтаксису використовуваних методів, а також контроль відповідності однойменних елементів і їх типів для різних типів діаграм.

При побудові DFD забезпечується контроль відповідності діаграм різних рівнів декомпозиції. Контроль за правильністю верхнього рівня DFD здійснюється за допомогою матриці списків подій (ELM). Для контролю за декомпозицією складених потоків даних використовується декілька варіантів їх опису: у вигляді діаграм структур даних (DSD) або в нотації БНФ (форма Бекуса-наура).

Для побудови SAD використовується розширена нотація DFD, що дає можливість вводити поняття процесорів, завдань і периферійних пристроїв, що забезпечує наочність проектних рішень.

При побудові моделі даних у вигляді ERD виконується її нормалізація і вводиться визначення фізичних імен елементів даних і таблиць, які використовуватимуться в процесі генерації фізичної схеми даних конкретної СУБД. Забезпечується можливість визначення альтернативних ключів суті і полів, складових додаткові точки входу в таблицю (поля індексів), і потужності відносин між суттю.

Наявність універсальної системи генерації коду, заснованої на специфікованих засобах доступу до репозиторія проекту, дозволяє підтримувати високий рівень виконання проектної дисципліни розробниками: жорсткий порядок формування моделей; жорстка структура і вміст документації; автоматична генерація початкових код програм і так далі - все це забезпечує підвищення якості і надійності тих, що розробляються ІС.



Для підготовки проектної документації можуть використовуватися видавничі системи FrameMaker, Interleaf або Word Perfect. Структура і склад проектної документації можуть бути настроєні відповідно до заданих стандартів. Налаштування виконується без зміни проектних рішень.

При розробці достатньо великою ІС вся система в цілому відповідає одному проекту як категорії Vantage Team Builder. Проект може бути декомпозований на ряд систем, кожна з яких відповідає деякій щодо автономній підсистемі ІС і розробляється незалежно від інших. Надалі системи проекту можуть бути інтегровані.

Процес проектування ІС з використанням Vantage Team Builder реалізується у вигляді 4-х послідовних фаз (стадій) - аналізу, архітектури, проектування і реалізації, при цьому закінчені результати кожної стадії повністю або частково переносяться (імпортуються) в наступну фазу. Всі діаграми, окрім ERD, перетворюються в інший тип або змінюють вигляд відповідно до особливостей поточної фази. Так, DFD перетворюється у фазі архітектури в SAD, DSD - в DTD. Після завершення імпорту логічний зв'язок з попередньою фазою розривається, тобто до діаграм можуть вноситися всі необхідні зміни.

#### *Взаємодія з іншими засобами*

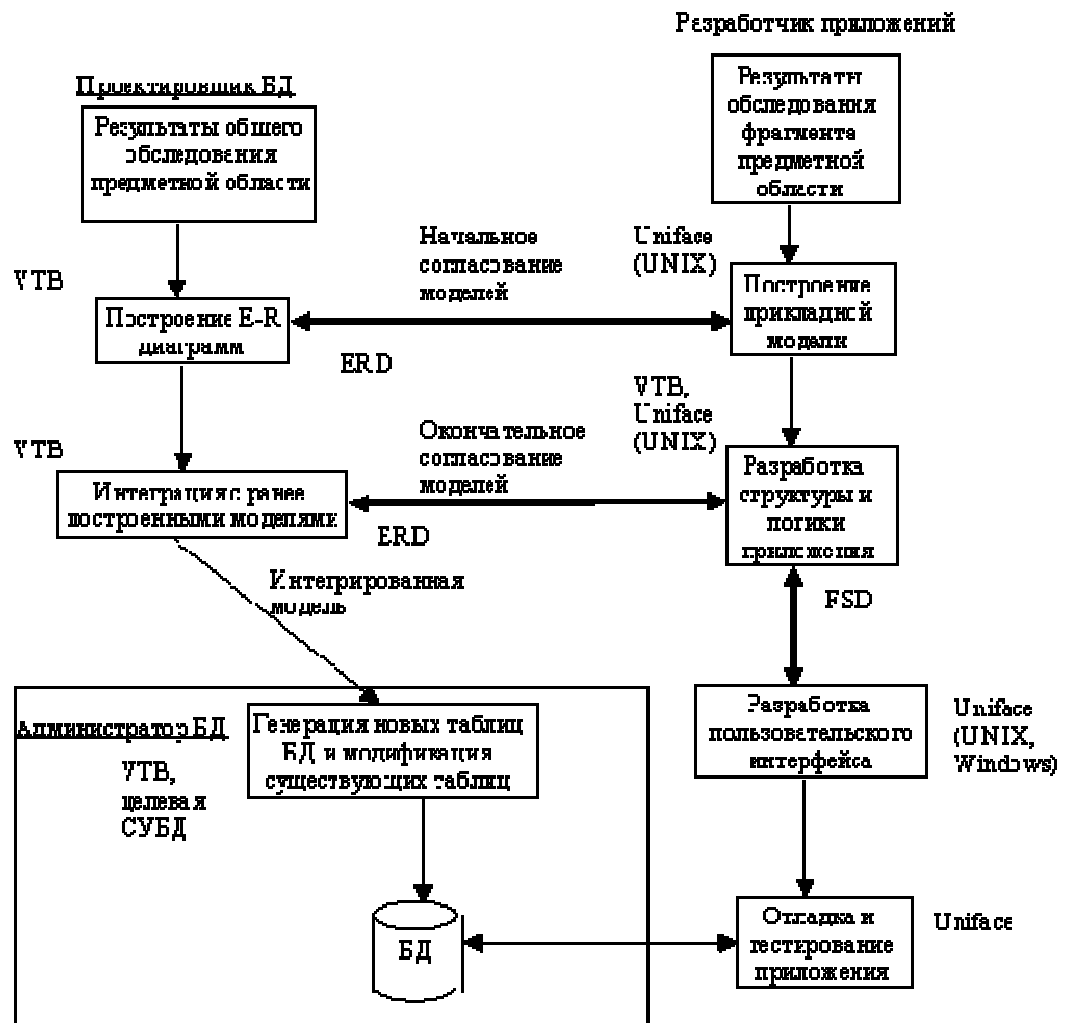
Конфігурація Vantage Team Builder for Uniface забезпечує сумісне використання двох систем в рамках єдиного технологічного середовища проектування, при цьому схеми БД (SQL-моделі) переносяться в репозиторій Uniface, і, навпаки, прикладні моделі, сформовані засобами Uniface, можуть бути перенесені в репозиторій Vantage Team Builder. Можливі розузгодження між репозиторіями двох систем усуваються за допомогою спеціальної утиліти. Розробка екранних форм в середовищі Uniface виконується на базі діаграм послідовностей форм (FSD) після імпорту SQL-моделі. Технологія розробки ІС на базі даної конфігурації показана на малюнку 5.1.

Структура репозиторія (що зберігається безпосередньо в цільовій СУБД) і інтерфейси Vantage Team Builder є відкритими, що в принципі дозволяє інтеграцію з будь-якими іншими засобами.

#### *Середовище функціонування*

Vantage Team Builder функціонує на всіх основних UNIX-платформах (Solaris, SCO UNIX, AIX, HP-UX) і VMS.

Vantage Team Builder можна використовувати в конфігурації "клієнт-сервер", при цьому база проектних даних може розташовуватися на сервері, а робочі місця розробників можуть бути клієнтами.



Мал. 5.1. Взаємодія Vantage Team Builder і Uniface

### 5.2.2. Uniface

Uniface 6.1 [15] - продукт фірми Compuware (США) - є середовищем розробки великомасштабних застосувань в архітектурі "клієнт-сервер" і має наступну компонентну архітектуру:

- Application Objects Repository (репозиторій об'єктів застосувань) містить метадані, автоматично використовувані рештою всіх компонентів впродовж життєвого циклу ІС (прикладні моделі, описи даних, бізнес-правил, екранних форм, глобальних об'єктів і шаблонів). Репозиторій може зберігатися в будь-якій з баз даних, підтримуваних Uniface;

- Application Model Manager підтримує прикладні моделі (E-R моделі), кожна з яких є підмножиною загальної схеми БД з погляду даного застосування, і включає відповідний графічний редактор;

- Rapid Application Builder - засіб швидкого створення екранних форм і звітів на базі об'єктів прикладної моделі. Воно включає графічний редактор форм, засобу прототипування, відладки, тестування і документування. Реалізований інтерфейс з різноманітними типами віконних елементів управління (Open Widget Interface) для існуючих графічних інтерфейсів - MS Windows (включаючи VBX), Motif, OS/2. Універсальний інтерфейс уявлення (Universal Presentation Interface) дозволяє використовувати одну і ту ж версію застосування в середовищі різних графічних інтерфейсів без зміни

програмної коди;

- Developer Services (служби розробника) - використовуються для підтримки крупних проектів і реалізують контроль версій (Uniface Version Control System), права доступу (розмежування повноважень), глобальні модифікації і так далі. Це забезпечує розробників засобами паралельного проектування, вхідного і вихідного контролю, пошуку, перегляду, підтримки і видачі звітів за даними системи контролю версій;

- Deployment Manager (управління розповсюдженням застосувань) - засоби, що дозволяють підготувати створене застосування для розповсюдження, встановлювати і супроводжувати його (при цьому платформа користувача може відрізнитися від платформи розробника). У їх склад входять мережеві драйвери і драйвери СУБД, сервер застосувань (полісервер), засоби розповсюдження застосувань і управління базами даних. Uniface підтримує інтерфейс практично зі всіма відомими програмно-апаратними платформами, СУБД, CASE-засобами, мережевими протоколами і менеджерами транзакцій;

- Personal Series (персональні засоби) - використовуються для створення складних запитів і звітів в графічній формі (Personal Query і Personal Access - PQ/PA), а також для перенесення даних в такі системи, як WinWord і Excel;

- Distributed Computing Manager - засіб інтеграції з менеджерами транзакцій Tuxedo, Encina, CICS, OSF DCE.

Оголошена в кінці 1996 р. версія Uniface 7 повністю підтримує розподілену модель обчислень і триланкову архітектуру "клієнт-сервер" (з можливістю зміни схеми декомпозиції застосувань на етапі виконання). Застосування, що створюються за допомогою Uniface 7, можуть виконуватися в гетерогенних операційних середовищах, що використовують різні мережеві протоколи, одночасно на декількох різнорідних платформах (у тому числі і в Internet).

У склад компонент Uniface 7 входять:

- Uniface Application Server - сервер застосувань для розподілених систем;

- WebEnabler - серверне ПЗ для експлуатації застосувань в Internet і Intranet;

- Name Server - серверне ПЗ, таке, що забезпечує використання розподілених прикладних ресурсів;

- PolyServer - засіб доступу до даним і інтеграції різних систем.

У список підтримуваних СУБД входять DB2, VSAM і IMS; PolyServer забезпечує також взаємодію з ОС MVS.

Середовище функціонування Uniface - все основні UNIX - платформи і MS Windows.

### **5.3. Designer/2000 + Developer/2000**

CASE-засобом Designer/2000 2.0 фірм ORACLE [23] є інтегрованим CASE-засобом, що забезпечує в сукупності із засобами розробки

застосувань Developer/2000 підтримку повного ЖЦ ПЗ для систем, що використовують СУБД ORACLE.

#### *Структура і функції*

Designer/2000 є сімейством методологій і що підтримують їх програмних продуктів. Базова методологія Designer/2000 (CASE\*Method) - структурна методологія проектування систем, що повністю охоплює всі етапи життєвого циклу ІС [8,9]. Відповідно до цієї методології на етапі планування визначаються цілі створення системи, пріоритети і обмеження, розробляється системна архітектура і план розробки ІС. В процесі аналізу будуються модель інформаційних потреб (діаграма "суть-зв'язок"), діаграма функціональної ієрархії (на основі функціональної декомпозиції ІС), матриця перехресних посилань і діаграма потоків даних.

На етапі проектування розробляється докладна архітектура ІС, проектується схема реляційної БД і програмні модулі, встановлюються перехресні посилання між компонентами ІС для аналізу їх взаємного впливу і контролю за змінами.

На етапі реалізації створюється БД, будуються прикладні системи, проводиться їх тестування, перевірка якості і відповідності вимогам користувачів. Створюється системна документація, матеріали для навчання і керівництва користувачів. На етапах експлуатації і супроводу аналізуються продуктивність і цілісність системи, виконується підтримка і, при необхідності, модифікація ІС;

Designer/2000 забезпечує графічний інтерфейс при розробці різних моделей (діаграм) наочної області. В процесі побудови моделей інформація про них заноситься в репозиторій. До складу Designer/2000 входять наступні компоненти:

- Repository Administrator - засоби управління репозиторієм (створення і видалення застосувань, управління доступом до даним з боку різних користувачів, експорт і імпорт даних);

- Repository Object Navigator - засоби доступу до репозиторія, що забезпечують багатовіконний об'єктно-орієнтований інтерфейс доступу до всіх елементів репозиторія;

- Process Modeller - засіб аналізу і моделювання ділової діяльності, що ґрунтується на концепціях реінжиніринга бизнес-процесов (BPR - Business Process Reengineering) і глобальної системи управління якістю (TQM - Total Quality Management);

- Systems Modeller - набір засобів побудови функціональних і інформаційних моделей проектованої ІС, що включає засоби для побудови діаграм "суть-зв'язок (Entity-Relationship Diagrammer)", діаграм функціональних ієрархій (Function Hierarchy Diagrammer), діаграм потоків даних (Data Flow Diagrammer) і засіб аналізу і модифікації зв'язків об'єктів репозиторія різних типів (Matrix Diagrammer);

- Systems Designer - набір засобів проектування ІС, що включає засіб побудови структури реляційної бази даних (Data Diagrammer), а також засоби

побудови діаграм, що відображають взаємодію з даними, ієрархію, структуру і логіку застосувань, що реалізовується процедурами, що зберігаються, на мові PL/SQL (Module Data Diagrammer, Module Structure Diagrammer і Module Logic Navigator);

- Server Generator - генератор описів об'єктів БД ORACLE (таблиць, індексів, ключів, послідовностей і так далі). Крім продуктів ORACLE, генерація і реінжиниринг БД може виконуватися для СУБД Informix, DB/2, Microsoft SQL Server, Sybase, а також для стандарту ANSI SQL DDL і баз даних, доступ до яких реалізується за допомогою ODBC;

- Forms Generator (генератор застосувань для ORACLE Forms). Застосування, що генеруються, включають різні екранні форми, засоби контролю даних, перевірки обмежень цілісності і автоматичні підказки. Подальша робота із застосуванням виконується в середовищі Developer/2000;

- Repository Reports - генератор стандартних звітів, що інтегрований з ORACLE Reports і дозволяє русифікувати звіти, а також змінювати структурне представлення інформації.

Репозиторієм Designer/2000 є сховище всіх проектних даних і може працювати в многопользовательском режимі, забезпечуючи паралельне оновлення інформації декількома розробниками. В процесі проектування автоматично підтримуються перехресні посилання між об'єктами словника і можуть генеруватися більше 70 стандартних звітів про модельовану наочну область. Фізичне середовище зберігання репозиторія - база даних ORACLE.

Генерація застосувань, крім продуктів ORACLE, виконується також для Visual Basic.

#### *Взаємодія з іншими засобами*

Designer/2000 можна інтегрувати з іншими засобами, використовуючи відкритий інтерфейс застосувань API (Application Programming Interface). Крім того, можна використовувати засіб ORACLE CASE Exchange для експорту/імпорту об'єктів репозиторія з метою обміну інформацією з іншими CASE-засобами.

Developer/2000 забезпечує розробку переносимих застосувань, що працюють в графічному середовищі Windows, Macintosh або Motif. У середовищі Windows інтеграція застосувань Developer/2000 з іншими засобами реалізується через механізм OLE і елементи VBX, що управляють. Взаємодія застосувань з іншими СУБД (DB/2, DB2/400, Rdb) реалізується за допомогою засобів ORACLE Client Adapter для ODBC, ORACLE Open Gateway і API.

#### *Середовище функціонування*

Середовище функціонування Designer/2000 і Developer/2000 - Windows 3.x, Windows 95, Windows NT.

### **5.4. Локальні засоби (ERwin, VPwin, S-Designor, CASE.Аналітик)**

ERwin - засіб концептуального моделювання БД [24], що використовує методологію IDEF1X (див. підрозділ 2.5). ERwin реалізує проектування схеми БД, генерацію її опису на мові цільової СУБД (ORACLE, Informix,

Ingres, Sybase, DB/2, Microsoft SQL Server, Progress і ін.) і реінжиниринг існуючої БД. ERwin випускається в декількох різних конфігураціях, орієнтованих на найбільш поширені засоби розробки застосувань 4GL. Версія ERwin/OPEN повністю сумісна із засобами розробки застосувань PowerBuilder і SQLWindows і дозволяє експортувати опис спроектованою БД безпосередньо в репозиторії даних засобів.

Для ряду засобів розробки застосувань (PowerBuilder, SQLWindows, Delphi, Visual Basic) виконується генерація форм і прототипів застосувань.

Мережева версія Erwin ModelMart забезпечує узгоджене проектування БД і застосувань в рамках робочої групи.

Erwin - засіб функціонального моделювання, що реалізовує методологію IDEF0 (див. підрозділ 2.2).

Можливі конфігурації і орієнтовна вартість засобів (без технічної підтримки) приведені в таблиці.

<b>Конфігурація</b>	<b>Вартість \$</b>
ERwin/ERX	3,295
Erwin	2,495
ERwin/ERX for PowerBuilder, Visual Basic, Progress	3,495
ERwin/ERX for Delphi	4,295
ERwin/Desktop for PowerBuilder, Visual Basic	495
ERwin/ERX for SQLWindows / Designer/2000 / Solaris	3,495 / 5,795 / 6,995
ModelMart 5 / 10 user	11,995 / 19,995
Erwin/OPEN for ModelMart	3,995

S-Designor 4.2 є CASE-засобом для проектування реляційних баз даних [25]. По своїх функціональних можливостях і вартості він близький до CASE-засобу ERwin, відрізняючись зовні використовуваною на діаграмах нотацією. S-Designor реалізує стандартну методологію моделювання даних і генерує опис БД для таких СУБД, як ORACLE, Informix, Ingres, Sybase, DB/2, Microsoft SQL Server і ін. Для існуючих систем виконується реінжиниринг БД.

S-Designor сумісний з поряд засобів розробки застосувань (PowerBuilder, Uniface, TeamWindows і ін.) і дозволяє експортувати опис БД в репозиторії даних засобів. Для PowerBuilder виконується також пряма генерація шаблонів застосувань.

CASE.Аналитик 1.1 [3] є практично єдиним в даний час конкурентоздатним вітчизняним CASE-засобом функціонального моделювання і реалізує побудову діаграм потоків даних відповідно до методології, описаної в підрозділі 2.3. Його основні функції:

- побудова і редагування DFD;
- аналіз діаграм і проектних специфікацій на повноту і

несуперечність;

- отримання різноманітних звітів за проектом;
- генерація макетів документів відповідно до вимог ГОСТ 19.XXX і 34.XXX.

Середовище функціонування: процесор - 386 і вище, основна пам'ять - 4 Мб, дискова пам'ять - 5 Мб, MS Windows 3.x або Windows 95.

Орієнтовна вартість:

- однопользовательская версія - 605 \$;
- многопользовательская версія (одне робоче місце) - 535 \$.

База даних проекту реалізована у форматі СУБД Paradox і є відкритою для доступу.

За допомогою окремого програмного продукту (Catherine) виконується обмін даними з CASE-засобом ERwin. При цьому з проекту, виконаного в CASE.Аналитике, експортується опис структур даних і накопичувачів даних, яке по певних правилах формує опис суті і їх атрибутів.

### **5.5. Об'єктно-орієнтовані CASE-засоби (Rational Rose)**

Rational Rose - CASE-засобом фірми Rational Software Corporation (США) - призначено для автоматизації етапів аналізу і проектування ПЗ, а також для генерації код на різних мовах і випуску проектної документації [21]. Rational Rose використовує синтез-методологію об'єктно-орієнтованого аналізу і проектування, засновану на підходах трьох провідних фахівців в даній області: Золить, Рамбо і Джекобсона. Розроблена ними універсальна нотація для моделювання об'єктів (UML - Unified Modeling Language) претендує на роль стандарту в області об'єктно-орієнтованого аналізу і проектування. Конкретний варіант Rational Rose визначається мовою, на якій генеруються коди програм (C++, Smalltalk, PowerBuilder, Ada, SQLWindows і ObjectPro). Основний варіант - Rational Rose/C++ - дозволяє розробляти проектну документацію у вигляді діаграм і специфікацій, а також генерувати програмні коди на C++. Крім того, Rational Rose містить засоби реінжиниринга програм, що забезпечують повторне використання програмних компонент в нових проектах.

#### *Структура і функції*

У основі роботи Rational Rose лежить побудова різного роду діаграм і специфікацій, що визначають логічну і фізичну структури моделі, її статичні і динамічні аспекти. До їх числа входять діаграми класів, станів, сценаріїв, модулів, процесів [21].

У складі Rational Rose можна виділити 6 основних структурних компонент: репозиторій, графічний інтерфейс користувача, засоби проглядання проекту (browser), засоби контролю проекту, засоби збору статистики і генератор документів. До них додаються генератор код (індивідуальний для кожної мови) і аналізатор для C++, забезпечуючий реінжиниринг - відновлення моделі проекту по початкових текстах програм.

Репозиторій є об'єктно-орієнтованою базою даних. Засоби перегляду забезпечують "навігацію" за проектом, зокрема, переміщення по ієрархіях

класів і підсистем, перемикання від одного вигляду діаграм до іншого і так далі Засоби контролю і збору статистики дають можливість знаходити і усувати помилки у міру розвитку проекту, а не після завершення його опису. Генератор звітів формує тексти вихідних документів на основі інформації, що міститься в репозиторії.

Засоби автоматичної генерації код програм на мові C++, використовуючи інформацію, що міститься в логічній і фізичній моделях проекту, формують файли заголовків і файли описів класів і об'єктів. Створюваний таким чином скелет програми може бути уточнений шляхом прямого програмування на мові C++. Аналізатор код C++ реалізований у вигляді окремого програмного модуля. Його призначення полягає в тому, щоб створювати модулі проектів у формі Rational Rose на основі інформації, що міститься у визначуваних користувачем початкових текстах на C++. В процесі роботи аналізатор здійснює контроль правильності початкових текстів і діагностику помилок. Модель, отримана в результаті його роботи, може цілком або фрагментарно використовуватися в різних проектах. Аналізатор володіє широкими можливостями настройки по входу і виходу. Наприклад, можна визначити типи початкових файлів, базовий компілятор, задати, яка інформація повинна бути включена у формовану модель і які елементи вихідної моделі слід виводити на екран. Таким чином, Rational Rose/C++ забезпечує можливість повторного використання програмних компонент.

В результаті розробки проекту за допомогою CASE-засобів Rational Rose формуються наступні документи:

- діаграми класів;
- діаграми станів;
- діаграми сценаріїв;
- діаграми модулів;
- діаграми процесів;
- специфікації класів, об'єктів, атрибутів і операцій
- заготовки текстів програм;
- модель програмної системи, що розробляється.

Останній з перерахованих документів є текстовим файлом, що містить всю необхідну інформацію про проект (зокрема необхідну для отримання всіх діаграм і специфікацій).

Тексти програм є заготовками для подальшої роботи програмістів. Вони формуються в робочому каталозі у вигляді файлів типів .h (заголовки, що містять описи класів) і .cpp (заготовки програм для методів). Система включає в програмні файли власні коментарі, які починаються з послідовності символів *///*###*. Склад інформації, що включається в програмні файли, визначається або за умовчанням, або по розсуду користувача. Надалі ці початкові тексти розвиваються програмістами в повноцінні програми.*

*Взаємодія з іншими засобами і організація групової роботи*

Rational Rose інтегрується із засобом PVCS для організації групової



роботи і управління проектом і із засобом SODA - для документування проектів. Інтеграція Rational Rose і SODA забезпечується засобами SODA.

Для організації групової роботи в Rational Rose можливе розбиття моделі на керовані підмоделі. Кожна з них незалежно зберігається на диску або завантажується в модель. Як підмоделі може виступати категорія класів або підсистема.

Для керованої підмоделі передбачені операції:

- завантаження підмоделі в пам'ять;
- вивантаження підмоделі з пам'яті;
- збереження підмоделі на диску у вигляді окремого файлу;
- установка захисту від модифікації;
- заміна підмоделі в пам'яті на нову.

Найефективніше групова робота організовується при інтеграції Rational Rose із спеціальними засобами управління конфігурацією і контролю версій (PVCS). В цьому випадку захист від модифікації встановлюється на всі керовані підмоделі, окрім тих, які виділені конкретному розробникові. В цьому випадку ознака захисту від запису встановлюється для файлів, які містять підмоделі, тому при прочитуванні "чужих" підмоделей захист їх від модифікації зберігається і випадкові дії виявляються неможливими.

*Середовище функціонування*

Rational Rose функціонує на різних платформах: IBM PC (у середовищі Windows), Sun SPARC stations (UNIX, Solaris, SUNOS), Hewlett-Packard (HP UX), IBM RS/6000 (AIX).

Для роботи системи необхідне виконання наступних вимог:

– Платформа Windows - процесор 80386SX або вище (рекомендується 80486), пам'ять 8Мб (рекомендується 12Мб), простір на диску 8Мб + 1-3Мб для однієї моделі.

– Платформа UNIX - пам'ять 32+(16\*число користувачів) Мб, простір на диску 30Мб + 20 при інсталяції + 1-3Мб для однієї моделі.

Сумісність по версіях забезпечується на рівні моделей.

## **5.6. Допоміжні засоби підтримки життєвого циклу ПЗ**

### **5.6.1. Засоби конфігураційного управління**

Мета конфігураційного управління (КУ) - забезпечити керованість і контрольованість процесів розробки і супроводу ПЗ. Для цього необхідна точна і достовірна інформація про стан ПЗ і його компонент в кожен момент часу, а також про всі передбачувані і виконані зміни.

Для вирішення завдань КУ застосовуються методи і засоби стани, що забезпечують ідентифікацію, компонент, облік номенклатури всіх компонент і модифікацій системи в цілому, контроль за змінами, що вносяться, в компоненти, структуру системи і її функції, а також координоване управління розвитком функцій і поліпшенням характеристик системи.

Найбільш поширеним засобом КУ є PVCS фірми Intersolv (США), що включає ряд самостійних продуктів: PVCS Version Manager, PVCS Tracker, PVCS Configuration Builder і PVCS Notify.

PVCS Version Manager [18] призначений для управління всіма компонентами проекту і ведення планомірної многоверсійної і багатоплатформеної розробки силами команди розробників в умовах однієї або декількох локальних мереж. Поняття "проект" трактується як сукупність файлів. В процесі роботи над проектом проміжний стан файлів періодично зберігається в архіві проекту, ведуться записи про час збереження, відповідність один одному декількох варіантів різних файлів проекту. Окрім цього, фіксуються імена розробників, відповідальних за той або інший файл, склад файлів проміжних версій проекту і ін. Це дозволяє повернутися при необхідності до якого-небудь з попередніх станів файлу (наприклад, при виявленні помилки, яку в даний момент важко виправити).

PVCS Version Manager призначений для використання в робочих групах. Система блокувань, реалізована в PVCS Version Manager дозволяє запобігти одночасному внесенню змін до одного і того ж файлу. В той же час, PVCS Version Manager дозволяє розробникам працювати з власними версіями загального файлу з напівавтоматичним дозволом конфліктів між ними.

Доступ до архівів PVCS Version Manager можливий не тільки через сам Version Manager, але і із понад 50 інструментальних засобів, зокрема MS Visual C і MS Visual Basic, Uniface, PowerBuilder, SQL Windows, JAM, Delphi, Paradox і ін.

Результатом роботи PVCS Version Manager є створений засобами файлової системи репозиторій, що зберігає в компактній формі всі робочі версії програмного продукту разом з необхідними коментарями і мітками.

PVCS Version Manager функціонує в середовищі MS Windows, Windows 95, Windows NT, OS/2, SUNOS, Solaris, HP-UX, AIX і SCO UNIX і може виконуватися на будь-якому персональному комп'ютері з процесором 80386 або вище, робочих станціях Sun, HP і IBM (RS-6000).

Іншим засобом конфігураційного управління є PVCS Tracker [19] - спеціалізована надбудова над офісною електронною поштою, призначена для обробки повідомлень про помилки в продукті, доставці їх виконавцям і контролю за виконання. Інтеграція з PVCS Version Manager дає можливість пов'язувати з повідомленнями ті або інші компоненти проекту. Звітні можливості PVCS Tracker включають безліч різновидів графіків і діаграм, що відображають стан проекту і процесу його відладки, зрізи по різних компонентах проекту, розробниках і тестувальниках. З їх допомогою можна наочно показати поточний стан роботи над проектом і її тимчасові тенденції.

Персонал, що працює з PVCS Tracker ділиться на п'ять груп залежно від їх обов'язків: користувачі, розробники, група тестування і контролю якості, група технічної підтримки і супроводу, управлінський персонал. Цим п'яти групам персоналу відповідають п'ять зумовлених груп PVCS Tracker:

- користувачі (Submitters) - мають обмежені має рацію на внесення зауважень і повідомлень про помилки в базу даних PVCS Tracker;
- розробники (Development Engineers) - мають право проводити основні операції з вимогами і зауваженнями в базі даних PVCS Tracker. Якщо

розробники діляться на підгрупи, то для кожної підгрупи можуть бути задані окремі списки прав доступу;

- тестувальники (Quality Engineers) - мають право проводити основні операції з вимогами і зауваженнями;

- супровід (Support Engineers) - мають право вносити будь-які зауваження, вимоги і рекомендації в базу даних, але не мають прав по розподілу робіт і зміні їх пріоритетності і термінів виконання;

- керівники (Managers) - мають право розподіляти роботи між виконавцями і ухвалювати рішення про їх належного виконання. Керівникам різних груп можуть задані різні права доступу до бази даних PVCS Tracker.

На додаток до цих п'яти зумовлених груп, існує група адміністратора бази даних і 11 додаткових груп, які можуть бути настроєні відповідно до специфічних посадових обов'язків співробітників, використовуючих PVCS Tracker.

Вимога або зауваження поступає в PVCS Tracker проходить чотири етапи обробки:

- реєстрація - внесення зауваження в базу даних;

- розподіл - призначення відповідального виконавця і термінів виконання;

- виконання - усунення зауваження, яке у свою чергу може викликати додаткові зауваження або вимоги на додаткові роботи;

- приймання - приймання робіт і зняття їх з контролю або напрям на доопрацювання.

Вимоги і зауваження, що поступають в базу даних PVCS Tracker оформляються у вигляді спеціальної форми, яка може містити до 18 полів вибору стандартних значень і до 12 довільних текстових рядків. При розробці форми слід визначити оптимальний набір інформації, характерний для всіх записів в базі даних.

Для отримання змістовної інформації про хід розробки PVCS Tracker дозволяє отримувати три типи статистичних звітів: частотні, тренди і діаграми розподілу.

Частотні звіти містять інформацію про частоту зауважень, що поступають, за одну годину тестування програмного продукту. Проте універсального частотного звіту не існує, оскільки на оцінку якості впливають тип методів тестування, серйозність виявлених помилок і значення дефектних модулів для функціонування всієї системи. Мале число фатальних помилок, що приводять до повної зупинки розробки, гірше за велике число зауважень до зовнішнього вигляду інтерфейсу користувача. Отже, частотні звіти повинні бути налаштовані на виявлення якого-небудь конкретного аспекту якості для того, щоб їх можна було використовувати для прогнозування закінчення робіт над проектом.

Тренди містять інформацію про зміни того або іншого показника в часі і характеризують стабільність і безперервність процесу розробки. Вони дозволяють відповісти на питання:

- чи встигає група розробників справлятися із зауваженнями, що поступають;
- чи поліпшується якість програмного продукту і яка динаміка цього процесу;
- як вплинуло те або інше рішення (збільшення числа розробників, введення ковзаючого графіка, впровадження нового методу тестування) на роботу групи і тому подібне

Діаграми розподілу - найбільш різноманітні і корисні для здійснення оперативного керівництва форми звітів. Вони дозволяють відповісти на питання: який метод тестування ефективніший, які модулі викликають найбільше число нарікань, хто з розробників краще справляється з конкретним типом завдань, чи немає перекосу в розподілі робіт між виконавцями, чи немає модулів, тестуванню яких було приділено недостатньо уваги і так далі

PVCS Tracker призначений для використання в робочих групах, об'єднаних в загальну мережу. В цьому випадку центральна база або проект PVCS Tracker знаходиться на загальнодоступному сервері мережі, доступ до якого реалізується за допомогою ODBC-драйверов, що входять до складу PVCS Tracker. Головною особливістю PVCS Tracker в порівнянні із звичайним застосуванням СУБД є його здатність автоматично повідомляти користувача про надходження що цікавить його або інформації, що відноситься до його компетенції, і гнучка система розподілу повноважень усередині робочої групи. При необхідності PVCS Tracker може використовувати для повідомлення видалених членів групи електронну пошту.

PVCS Tracker підтримує групову роботу в локальних мережах і взаємодіє з СУБД dBase, ORACLE, SQL Server і SYBASE за допомогою ODBC.

PVCS Tracker може бути інтегрований з будь-якою системою електронної пошти, що підтримує стандарти VIM, MAPI або SMTP.

PVCS Version Manager і PVCS Tracker оточені допоміжними компонентами: PVCS Configuration Builder і PVCS Notify.

PVCS Configuration Builder призначений для збірки остаточного продукту з компонент проекту. PVCS Configuration Builder дозволяє описувати процес збірки як на стандартній мові MAKE, так і на власній внутрішній мові, що має істотно великі можливості. PVCS Configuration Builder дозволяє здійснювати збірку програмного продукту на підставі файлів, що зберігаються в репозиторії PVCS Version Manager.

Звичайна процедура збірки програмного продукту за допомогою PVCS Configuration Builder складається з трьох кроків:

- будується файл залежностей між початковими модулями;
- до отриманого файлу вносяться зміни з метою його настройки і оптимізації;
- здійснюється збірка програмного продукту з початкових модулів.

Результатом роботи PVCS Configuration Builder є спеціальний файл, що описує оптимальний алгоритм збірки програмного продукту, побудований на основі аналізу дерева залежностей між початковими модулями.

PVCS Notify забезпечує автоматичну розсилку повідомлень про помилки з бази даних пакету PVCS Tracker по робочих станціях призначення. При цьому використовується офісна система електронної пошти cc:Mail або Microsoft Mail. PVCS Notify розширює можливості PVCS Tracker і використовується тільки спільно з ним.

PVCS Notify настраюється з середовища PVCS Tracker. Налаштування включає визначення інтервалу часу, через який PVCS Notify перевіряє вміст бази даних, визначення критеріїв відбору записів для розсилки повідомлень, визначення списків адрес для розсилки. Після налаштування PVCS Notify починає роботу в автономному режимі, автоматично розсилаючи повідомлення про зміни в базі даних PVCS Tracker.

PVCS Notify призначений для використання у великих робочих групах, частина членів яких хоча і доступна тільки через засоби електронної пошти, проте повинна мати оперативну інформацію про вимоги на зміну програмного продукту, зауваження, помилки, ході і результатах його тестування.

Результатом роботи PVCS Notify є оформлені відповідно до одного із стандартів поштової повідомлення, готові для розсилки за допомогою системи електронної пошти.

#### **5.6.2. Засоби документування**

Для створення документації в процесі розробки ІС використовуються різноманітні засоби формування звітів, а також компоненти видавничих систем. Зазвичай засоби документування вбудовані в конкретні CASE-засоби. Виключенням є деякі пакети, що надають додатковий сервіс при документуванні. З них найактивніше використовується SODA (Software Document Automation).

Продукт SODA призначений для автоматизації розробки проектною документації на всіх фазах ЖЦ ПЗ. Він дозволяє автоматично витягувати різноманітну інформацію, що отримується на різних стадіях розробки проекту, і включати її у вихідні документи. При цьому контролюється відповідність документації проекту, взаємозв'язок документів, забезпечується їх своєчасне оновлення. Результуюча документація автоматично формується з безлічі джерел, число яких не обмежене.

SODA не залежить від вживаних інструментальних засобів. Зв'язок із застосуваннями здійснюється через стандартний програмний інтерфейс API. Перехід на нові інструментальні засоби не спричиняє за собою додаткових витрат по документуванню проекту.

SODA містить набір шаблонів документів, визначуваних стандартом на програмне забезпечення DOD 2167A. На їх основі можна без спеціального програмування створювати нові форми документів, визначувані користувачами.

Пакет включає графічний редактор для підготовки шаблонів

документів. Він дозволяє задавати необхідний стиль, фон, шрифт, визначати розташування заголовків, резервувати місця, де розміщуватиметься витягувана з різноманітних джерел інформація. Зміни автоматично вносяться тільки до тих частин документація, на яку вони вплинули в програмі. Це скорочує час підготовки документації за рахунок відмови від регенерації всієї документації.

SODA реалізована на базі видавничої системи FrameBuilder і надає повний набір засобів по редагуванню і верстці документації, що випускається. Різні версії документації можуть бути для наочності відмічені своїми відмітними ознаками. У системі створюються таблиці вимог до проекту, по яких можна прослідкувати, як реалізуються ці вимоги. Різні види документація, супроводжуюча різні етапи ЖЦ, зв'язана між собою, і можна прослідкувати стан проекту від первинних вимог до аналізу, проектування, кодування і тестування програмного продукту.

Підсумковим результатом роботи системи SODA є готовий документ (або книга). Документ може зберігатися у файлі формату SODA (Frame Builder), який виходить в результаті генерації документа. Вивід на друк цього документа (або його частини) можливий з системи SODA.

Середовище функціонування SODA - ОС типу UNIX на робочих станціях Sun SPARCstation, IBM RISC System/6000 або Hewlett Packard HP 9000 700/800.

SODA вимагає принаймні 32 МВ оперативної пам'яті, 100-300 МВ для установки і 64 МВ робочого простору на диску.

### **5.6.3. Засоби тестування**

Під тестуванням розуміється процес виконання програми з метою виявлення помилок. Регресійне тестування - це тестування, що проводиться після удосконалення функцій програми або внесення до неї змін.

Один з найбільш розвинених засобів тестування QA (нова назва - Quality Works) [20] є інтегрованим, багатоплатформеним середовищем для розробки автоматизованих тестів будь-якого рівня, включаючи тести регресії для застосувань з графічним інтерфейсом користувача.

QA дозволяє починати тестування на будь-якій фазі ЖЦ, планувати і управляти процесом тестування, відображати зміни в застосуванні і повторно використовувати тести більш ніж 25 різних платформ.

Основними компонентами QA є:

- QA Partner - середовище для розробки, компіляції і виконання тестів;
- QA Planner - модуль для розробки планів тестування і обробки результатів. Для створення і виконання тестів в процесі роботи QA Planner викликається QA Partner;
- Agent - модуль, що підтримує роботу в мережі.
- Процес тестування складається з наступних етапів:
  - створення плану тестування;
  - пов'язання плану з тестами;

- позначка і виконання тестів;
- отримання звітів про тестування і управління результатами.

Створення тестового плану в QA Planner включає складання схеми тестових вимог і виділення рівнів деталізації. Для цього необхідно визначити все, що повинне бути протестоване, підготувати функціональну декомпозицію застосування, оцінити, скільки тестів необхідно для кожної функції і характеристики, визначити, скільки з них буде реалізовано залежно від доступних ресурсів і часу. Ця інформація використовується для створення схеми тестових вимог.

Для пов'язання плану з тестами необхідно створити пропозиції (скрипти), що управляють, на спеціальній мові 4Test і тести, які виконують вимоги плану, і зв'язати компоненти будь-яким способом. Для уникнення перевантаженості тестів використовують управління тестовими даними.

При виконанні плану результати записуються у форматі, схожому на план. Всі результати пов'язані з планом. Є можливість проглянути або приховати загальну інформацію про виконання, злити файли результатів, розмітити невдалі тести, порівняти результати попереднього виконання тестів, виконати або відмінити звіт.

Одним з атрибутів тесту є ім'я його розробника, що дозволяє при необхідності виконувати тести, створені конкретним розробником.

Комплекс QA займає на жорсткому диску не більш 21MB. Підтримувані платформи: Windows 3.x, Windows 95, Windows NT, OS/2, Macintosh, VMS, HP-UX, AIX, Solaris.

### **5.7. Приклади комплексів CASE-засобів**

На закінчення приведемо приклади комплексів CASE-засобів що забезпечують підтримку повного ЖЦ ПЗ. Тут хотілося б ще раз відзначити недоцільність порівняння окремо узятих CASE-засобів, оскільки жодне з них не вирішує в цілому всі проблеми створення і супроводу ПЗ. Це підтверджується також повним набором критеріїв оцінки і вибору, які зачіпають всі етапи ЖЦ ПЗ. Порівнюватися можуть комплекси методологічно і технологічно узгоджених інструментальних засобів, що підтримують повний ЖЦ ПЗ і забезпечені необхідною технічною і методичною підтримкою з боку фірм-постачальників. На думку автора, на сьогоднішній день найбільш розвиненим зі всіх комплексів, що поставляються в Росії, такого роду є комплекс технологій і інструментальних засобів створення ІС, заснований на методології і технології DATARUN. До складу комплексу входять наступні інструментальні засоби:

- CASE-засіб Silverrun;
- засіб розробки застосувань JAM;
- міст SILVERRUN-RDM <-> JAM;
- комплекс засобів тестування QA;
- менеджер транзакцій Tuxedo;
- комплекс засобів планування і управління проектом SE Companion;
- комплекс засобів конфігураційного управління PVCS;

- об'єктно-орієнтований CASE-засіб Rational Rose;
  - засіб документування SODA.
- Прикладами інших подібних комплексів є:
- Vantage Team Builder for Uniface + Uniface (фірми "DataX/Florin" і "ЩІК");
  - комплекс засобів, що поставляються і використовуються фірмою "ФОРС":
    - CASE-засіб Designer/2000 (основне), ERwin, Vrwin і Oowin (альтернативні);
    - засоби розробки застосувань Developer/2000, ORACLE Power Objects (основні) і Usoft Developer (альтернативне);
    - засіб настройки і оптимізації EXPLAINSOL (Platinum);
    - засоби адміністрування і супроводу SQLWatch, DBVision, SQL Spy, TSReorg і ін. (Platinum);
    - засіб документування ORACLE Book.
  - комплекс засобів на основі продуктів фірми CENTURA:
    - CASE-засобів ERwin, Vrwin і Oowin (об'єктно-орієнтований аналіз);
    - засоби розробки застосувань SQLWindows і TeamWindows;
    - засіб тестування і оптимізації застосувань "клієнт-сервер" SQLBench (ARC);
    - засоби експлуатації і супроводу Quest і Crystal Reports.



## **6. Моделі управління підприємствами в інформаційних системах**

### **6.1. Загальна характеристика систем автоматизації управління ресурсами підприємства**

Організаційна структура управління підприємством визначає функції управління структурних підрозділів і регламентує схему інформаційних потоків системи управління, рівні прийняття управлінських рішень. Організаційна структура управління зазвичай характеризується кількістю ланок управління і слугує основою організаційних комунікацій. Типовими організаційними структурами управління є:

- лінійно-функціональна структура, що закріплює за підрозділом обмежені функції управління;
- дивізійна структура на основі бізнес-процесів, що закріплює за підрозділом функції повного управлінського циклу;
- матрична структура, що об'єднує функції лінійно-функціональної і дивізійної структур.

На сьогодні лінійно-функціональна структура використовується для дрібних і середніх фірм. Для великих компаній із середини 80-х років використовується дивізійна структура управління. Це пов'язано з децентралізацією керування, наданням оперативно-виробничої і фінансової самостійності структурним одиницям. Дивізійні структури - результат розвитку підприємств і появи самостійних підрозділів, розташованих у різних географічних регіонах, як результат диверсифікованості бізнесу.

На підприємствах з лінійно-функціональною організаційною структурою застосовуються адміністративно-централізовані методи управління. У системі керування виникають і циркулюють великі вертикальні інформаційні потоки. Горизонтальні зв'язки між окремими управлінськими функціями реалізуються на верхніх рівнях ієрархії, де приймаються стратегічні рішення. Управлінський персонал використовує функціонально орієнтовані інформаційні системи збору й обробки інформації. При дивізійній структурі організації бізнес-процесів виникає проблема централізації функцій керування в масштабі підприємства й агрегування інформації для прийняття стратегічних управлінських рішень. Матрична організаційна структура керування є симбіозом двох організаційних структур і може бути ефективною в умовах добре функціонуючих організаційних одиниць і використання сучасних ІТ.

Фінансову структуру підприємства утворюють центри фінансового обліку і відповідальності. Ця структура є основою фінансового планування підприємства. У складі фінансової структури виділяють центри:

- фінансового обліку - підрозділ, функції якого впливають на прибутковість діяльності підприємства;
- фінансової відповідальності - підрозділ, метою якого є максимізація прибутку;
- профіт-центр - підрозділ, діяльність якого спрямована на реалізацію бізнес-проекту й одержання прибутку;

– венчур-центр - підрозділ, що організує новий бізнес-проект, прибуток від якого очікується в майбутньому;

– витрат - підрозділ, що забезпечує підтримку й обслуговування венчур- і профіт-центрів.

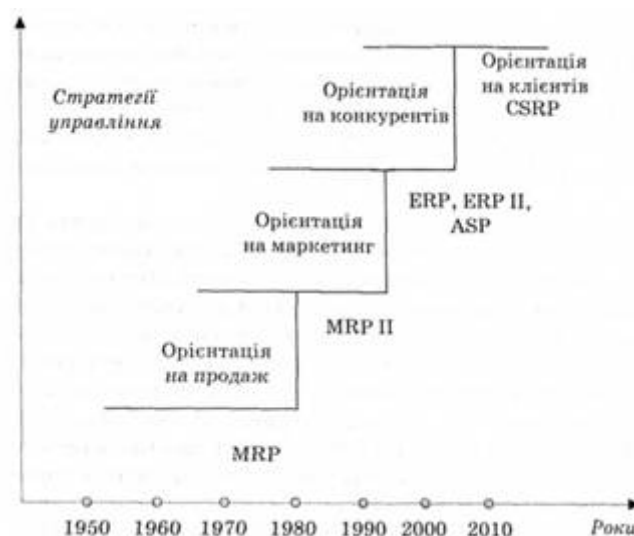
Зміцнення центрів фінансової відповідальності пов'язане з виведенням їх зі структури компанії, перетворенням їх у холдинги або створенням інших форм групової взаємодії (фінансово-промислові, промислові і комерційні групи, синдикати, концерни, картелі).

З урахуванням організаційної і фінансової структури підприємства, зовнішніх і внутрішніх економічних умов вибираються методи управління діяльністю підприємства, що забезпечують досягнення мети бізнес-процесів.

Початок масового використання ІС для промислового виробництва можна віднести до середини 60-х років, коли зі зниженням вартості комп'ютерів ІС почали застосовуватися для планування виробничих процесів підприємств. Дослідження взаємозв'язку технології й організаційних структур вперше здійснили Чепл і Сейлс, які стверджували, що тип технології має бути основним критерієм при проектуванні організаційної структури. На рис. 6.1 відображено еволюцію бізнес-стратегій та відповідних концепцій побудови ІС планування й управління ресурсами підприємства.

Починаючи із середини 50-х до середини 70-х років зарубіжні компанії взяли на озброєння економічну стратегію, що дістала назву стратегії орієнтації на продаж. Суть стратегії полягала у плануванні конкретного обсягу продукції за конкретний період часу. Процес розподілу продукції в транснаціональних корпораціях потребував обліку та управління матеріальними потоками, що стало основною причиною для створення в 60-х роках програмних комплексів управління потребами в матеріалах.

Концепція управління матеріальними ресурсами отримала назву MRP (Material Resource Planning - планування виробничих ресурсів), а відповідні ІС - MRP-систем. MRP - це методи управління промисловим підприємством в умовах конкурентної ринкової економіки.



**Рис. 6.1.** Еволюція бізнес-стратегій ІС підприємства

У 70-х роках концепція орієнтації на продаж змінилася стратегією орієнтації на маркетинг, або на споживача. Тому з'явилася нова концепція управління виробничими ресурсами - MRP II, основною ідеєю якої було управління виробництвом на всіх фазах - від постачання сировини до відвантаження готової продукції споживачам. MRP II описує методологію, що ефективно управляє всіма ресурсами підприємства, забезпечуючи оптимальні рішення проблем планування діяльності підприємства в натуральних одиницях, фінансове планування в грошовому еквіваленті.

ERP (Enterprise Resource Planning) - управління ресурсами (матеріальними, фінансовими, трудовими) у межах корпорації. Стандарт CSRP орієнтований на керування зовнішніми

елементами виробничого ланцюжка підприємства, орієнтується на клієнта, визначається успішною реалізацією електронних технологій у бізнесі, що створюють необхідну базу для загальної динаміки ринкових процесів і вимог споживачів. Саме зміни в цих сферах дають змогу стверджувати, що економіка вступає в нову епоху - інформаційну масово-персоніфіковану.

Головна вимога цієї епохи - необхідність поєднання індивідуальних переваг споживача з ефективним виробництвом і системою планування комерційної діяльності. Для реалізації цієї вимоги стали запроваджуватися нові системи управління ресурсами підприємств, що включають управління закупівлями, виробництвом, збутом, складськими запасами, трудовими і фінансовими ресурсами, конструкторськими розробками у режимі реального часу. Системи управління ресурсами підприємства в новій економіці можуть стати основою ефективних комунікацій. Запровадження цих систем дає змогу оптимізувати економічні процеси підприємства і зробити можливим максимально повне використання переваг систем електронної комерції.

Успішне зростання *інформаційної економіки* сприяє запровадженню різних систем автоматизації управління ресурсами підприємства (MRP II, ERP, ERP II, CRM, SCM, SRM, CSRP тощо) (табл. 6.1).

**Таблиця 6.1.** Стандарти управління ІС

Назва стандарту управління	Опис
MRP Material Requirements Planning	Планування матеріалів й управління ними для виробництва
MRP II Manufacturing Resource Planning	Управління корпоративними ресурсами. До властивостей MRPII додалося також управління фінансовими ресурсами, маркетинг
ERP Enterprise Resource Planning	ЕІР-концепція - перша, яка спрямована на управління бізнесом, а не тільки виробництвом.

Назва стандарту управління	Опис
	Автоматизоване планування потреб у сировині і матеріалах. Планування й управління всіма виробничими ресурсами підприємства: сировиною, матеріалами, устаткуванням, трудовитратами
SCM Supply Chain Management	Управління відносинами з постачальниками
SRM Storage Resource Management	Управління зберіганням, що охоплює моніторинг стану, конфігурації, доступності, продуктивності і використання ресурсів, а також виконує генерацію звітів і розсилку попереджень. Рідко виділяється в самостійний клас управління. Входить в EPR
CRM Customer Relationships Management	Управління відносинами із замовниками. Відстежує історію розвитку взаєностосунків, координує багатобічні зв'язки, централізовано управляє продажем і клієнт-орієнтованим маркетингом'
CSRP Customer Synchronized Resource Pelanning	Управління, орієнтоване на взаємодію з клієнтами. Включає отримання замовлень, розробку планів, проектів і завдань, техпід-тримку. Практично, SCRP = ERP + CRM
ERP II	Нова ревізія концепції ERP. Можна вважати що, ERP II = ERP + CRM + SCM
ISO9000	Група стандартів щодо системи менеджменту якості, в тому числі якості управління, якості управлінських систем
HR Human Resources	Управління персоналом компанії: кадровий облік, облік робочого часу, розрахунок зарплати, навчання, прийняття персоналу, аналіз ефективності використання робочої сили

## 6.2. MRP-система (планування матеріальних ресурсів)

MRP – концепція планування потреби виробництва в матеріальних ресурсах, яка для визначення потреб використовує інформацію про структуру і технологію виробництва кінцевого продукту, календарний план виробництва, дані складських запасів, договорів поставки матеріалів і комплектуючих тощо.

MRP-система – інтегрована електронна інформаційна система управління, що реалізовує концепцію MRP. Розробка теорії MRP-систем здійснювалася ще на початку 60-х років. Нині ці системи присутні практично у всіх інтегрованих інформаційних системах управління підприємствами.

Основна мета MRP-систем полягає у тому, що будь-яка облікова одиниця ресурсів (товарно-матеріальних цінностей) має бути наявною в потрібний час і в потрібному місці. Принципи функціонування MRP-системи базуються на формуванні, контролі і за необхідності відбувається коригування параметрів надходження матеріальних ресурсів у такий спосіб, щоб всі матеріали, необхідні для виробництва, надходили до моменту їх споживання. Ці технології пов'язують окремі підрозділи підприємства, що займаються питаннями управління постачанням, виробничими процесами, обслуговуванням складів.

На початку 60-х років у зв'язку із поширенням електронних обчислювальних систем виникла ідея використовувати їх для планування діяльності підприємства, зокрема планування виробничих процесів. Необхідність планування була зумовлена тим, що затримки у процесі виробництва пов'язані із запізнюванням надходження окремих комплектуючих, що призводило до зниження ефективності виробництва, а передчасне надходження комплектуючих призводило до збільшення витрат на утримання складського господарства. Методика MRP якраз і займалася вирішенням подібних проблем. Практична реалізація MRP-системи дала змогу оптимально регулювати поставки матеріалів і комплектуючих для виробничого процесу, контролюючи їх витрати у виробництві і складські запаси (рис. 6.2).



**Рис. 6.2.** Структура MRP-системи

Методи планування MRP враховують інформацію про склад виробу, стан складів і незавершеного виробництва, а також замовлень на поставку готової продукції та планів-графіків виробництва і полягають у такому:

– замовлення упорядковуються, наприклад, за пріоритетами або термінами відвантаження;

– формується план-графік виробництва. Зазвичай він створюється за групами продукції і може бути використаний для планування завантаження виробничих потужностей;

– для кожного виробу, що потрапив у план-графік виробництва, склад виробу деталізується до рівня заготовок, напівфабрикатів, вузлів і комплектуючих;

– відповідно до плану-графіка виробництва визначається графік випуску вузлів і напівфабрикатів, а також оцінюється потреба в матеріалах і комплектуючих і призначаються терміни їх поставки у виробничі підрозділи.

Процес функціонування MRP-системи включає автоматичне формування замовлень на закупівлю і внутрішнє виробництво необхідних комплектуючих. Іншими словами, MRP-система оптимізує час поставки комплектуючих, зменшуючи виробничі витрати і підвищуючи ефективність виробничої діяльності. Основними перевагами використання подібної системи є гарантія своєчасного надходження матеріалів і комплектуючих, оптимізація складських запасів, зменшення виробничого браку при збірці готової продукції, що виникає внаслідок використання невідповідних комплектуючих, впорядкування виробництва внаслідок контролю статусу кожного матеріалу, що дозволяє відстежувати весь цикл його використання/починаючи від замовлення на даний матеріал до його використання в готовому виробі. Завдяки цьому досягається ефективність виробничого обліку.

Опишемо основні інформаційні елементи MRP-системи і операції з ними (рис. 6.2). **Опис стану матеріалів** (inventory status file) - це повна інформація про всі матеріали і комплектуючі, необхідні для виробництва кінцевого продукту. У цьому елементі вказаний статус кожного матеріалу, що визначає його положення (у виробництві, на складі, в поточних або планованих замовленнях), обсяг запасів, ціни, можливі затримки поставок, реквізити постачальників тощо. Вся інформація має бути одержана окремо за кожним матеріалом, що бере участь у виробничому процесі.

**План-графік виробництва** (master production schedule) - оптимізований графік виробництва готової продукції на запланований період. У разі недоступності деяких матеріалів MRP-система інформує про необхідність внесення в програму виробництва відповідного коригування.

**Специфікація виробу** (bills of material file) - список матеріалів з їх кількісними й якісними характеристиками, що необхідні для виробництва кінцевого продукту.

Цикл роботи MRP-програми складається з таких основних етапів:

1) на основі аналізу плану-графіка виробництва і специфікацій виробів, що знаходяться у виробництві, визначається повна потреба в матеріалах і комплектуючих на запланований період;

2) відповідно до даних про повну потребу і поточний стан матеріальних запасів для кожного періоду часу і для кожного матеріалу обчислюють чисту

потребу, тобто повну потребу за вирахуванням матеріалу, що знаходиться у виробництві, його страхового запасу і резерву;

3) на основі чистої потреби в матеріалах, використовуючи економіко-математичні методи управління запасами, складають графік поставки, в якому коректуються всі раніше створені заявки і формуються нові.

Основний результат роботи MRP-системи - це план замовлень (planned order schedule), що визначає графік поставки необхідних для виконання виробничої програми матеріалів і комплектуючих.

Крім плану замовлень MRP-система формує спеціальні звіти:

1) звіт про вузькі місця планування (exception report), що призначений для інформування про вузькі місця в графіку поставки, які вимагають особливої уваги і в яких може виникнути необхідність зовнішнього втручання. Вузькі місця визначаються шляхом моделювання розвитку ситуації у разі най-вірогідніших порушень поставок;

2) виконавчий звіт (performance report) – перелік невирішених критичних ситуацій, виявлених у процесі формування плану поставок, кожна з яких призводить до неможливості складання плану поставок, який забезпечив би виконання заданої виробничої програми;

3) звіт про прогнози (planning report) – інформація, що використовується для складання прогнозів перспективної зміни обсягів і характеристик продукції, одержана в результаті аналізу звіту про продаж і поточний стан виробничого процесу. Звіт про прогнози може також використовуватися для довгострокового планування потреб у матеріалах.

Таким чином, використання MRP-системи для планування виробничих потреб допомагає підвищити надійність забезпечення необхідними матеріалами виробничого процесу й оптимізувати час поставки матеріалів, значно знижуючи складські витрати.

Історично MRP призначалася для контролю над запасами і їх поповненням. Згодом технологія MRP була розширена, доповнена плануванням потреб у потужностях (CRP - Capacity Resources Planning), фінансовим плануванням (FRP - Finance Requirements Planning), плануванням кадрових ресурсів і завершенням всього ланцюжка планування у рамках системи MRP II.

Спочатку MRP-системи формували на основі затвердженої виробничої програми план замовлень на певний період. Для збільшення ефективності планування в кінці 70-х років Олівер Уайт і Джордж Плосл запропонували ідею відтворення замкненого циклу (closed loop) в MRP-системах, що охоплює весь виробничий процес від вивчення попиту до реалізації готової продукції. Термін "замкнений цикл" відображає основну особливість модифікованої системи, яка полягає у тому, що створювані в процесі її роботи матеріали аналізуються і враховуються на всіх етапах планування, змінюючи за необхідності програму виробництва, а отже, і план замовлень. Тобто додаткові функції підтримують зворотний зв'язок, забезпечуючи гнучкість планування стосовно зовнішніх чинників, таких як рівень попиту, стан справ у постачальників тощо. Саме процеси прогнозування, планування

і контролю виробництва із замкненим циклом планування спонукали до появи концепції MRP II.

### **6.3. MRP II-система (планування виробничих ресурсів)**

*Manufacture Resources Planning II* (MRP II - планування виробничих ресурсів) - концепція управління виробничим підприємством, що ґрунтується на взаємопов'язаному плануванні виробничих потужностей, потребі в матеріалах, фінансах і кадрах.

MRP II-методологія підтримує оперативне планування продукції і матеріалів у натуральних одиницях, фінансове планування у вартісних одиницях.

MRP II-система - інтегрована електронна інформаційна система управління, що реалізовує концепцію MRP II. Ця система, на відміну від MRP-системи, має можливості прогнозування і моделювання.

Завдання інформаційних систем класу MRP II - оптимальне управління потоками матеріалів (сировини), напівфабрикатів і готових виробів. Ці проблеми розв'язуються шляхом інтеграції всіх основних процесів, що реалізуються підприємством: постачання, управління запасами, виробництво, прямий продаж і дистрибуція.

Системи класу MRP II здійснюють:

- планування діяльності підприємства й управління від організації закупівлі до відвантаження товару споживачу;
- оптимізацію матеріальних і фінансових потоків;
- систематизацію управління інформацією: інформація заноситься в систему тільки один раз в місці виникнення і використовується всіма зацікавленими користувачами.

MRP II-системи дозволяють досягти узгодженості роботи різних підрозділів, знижуючи при цьому адміністративні витрати. Ці системи підвищують ефективність управління і прийняття управлінських рішень. Використання MRP II-систем дає змогу досягти конкурентних переваг шляхом оптимізації більшості господарських процесів підприємства.

Стандарт MRP II розроблено в США і підтримується Американським співтовариством управління виробництвом і запасами (American Production and Inventory Control Society, APICS). APICS регулярно публікує стандарт MRP II Standart System, в якому визначаються основні вимоги до інформаційних виробничих систем.

Стандарт MRP II містить опис 16 груп функцій:

- 1) планування продажу і виробництва (sales and operation planning);
- 2) управління попитом (demand management);
- 3) складання плану виробництва (master production scheduling);
- 4) планування матеріальних потреб (material requirement planning);
- 5) специфікації продуктів (bill of materials);
- 6) управління складом (inventory transaction subsystem);
- 7) планові поставки (scheduled receipts subsystem);
- 8) управління на рівні виробничого цеху (shop flow control);



- 9) планування виробничих потужностей (capacity requirement planning);
- 10) контроль входу/виходу (input/output control);
- 11) матеріальне технічне постачання (purchasing);
- 12) планування ресурсів розподілу (distribution resource planning);
- 13) планування і контроль виробничих операцій (tooling planning and control);
- 14) управління фінансами (financial planning);
- 15) моделювання (simulation);
- 16) оцінка результатів діяльності (performance measurement).

Технологія MRP II передбачає взаємне узгодження ресурсів у цілому на всьому підприємстві. Можливості інтегрованих систем стандарту MRP II:

- отримання оперативної інформації про поточні результати діяльності підприємства як у цілому, так і за окремими замовленнями, видами ресурсів, виконанням планів;
- повне планування діяльності підприємства (довго- та короткострокове), що здійснюється на основі оперативної інформації, для досягнення максимальної ефективності у використанні виробничих потужностей, всіх видів ресурсів і повного задоволення потреб замовників;
- оптимізація фінансових і матеріальних потоків;
- оптимізація обсягів матеріальних ресурсів на складах;
- значне скорочення невиробничих витрат.

Система планування в MRP II складається з кількох взаємопов'язаних планів: об'ємно-календарного - головного плану-графіка виробництва, плану продажу, випуску готової продукції, потреби в матеріалах, потреби у виробничих потужностях, фінансового плану.

В основі технології MRP II встановлено ієрархію планів (рис. 6.3). Плани нижніх рівнів визначаються характеристиками планів вищих рівнів. Планування здійснюється ітерацій -но, тобто формування плану нижчого рівня може вплинути на перегляд планів вищого рівня. Отже, виконується координація попиту і пропозиції ресурсів на всіх рівнях планування.

Зміна зовнішніх чинників, що впливає на план нижчого рівня, аналізується системою в цілому, забезпечуючи коректність планування і швидку адаптацію до змін на ринку. Швидка адаптація із замкненим циклом планування - це важлива властивість сучасних систем планування, оскільки значна кількість виробників виробляє продукцію з коротким життєвим циклом, що призводить до необхідності регулярного перегляду планів усіх рівнів.

Головний план-графік виробництва - це центральний стержень оперативного управління в системі MRP II. У ньому відображається інформація, що поступає з виробництва і відділу постачання, на нього накладають обмеження планові цифри щодо ресурсів і фінансових ресурсів. Він містить інформацію, на основі якої можна укласти договори поставки із замовниками, прогнозувати прибутки й управляти збутом.

У системах MRP II пропонується спеціальний інструментарій формування фінансового плану і складання бюджетних кошторисів, прогнозування й управління рухом грошових потоків. При фінансовому аналізі, що проводиться в рамках MRP II, не враховуються непрямі витрати (накладні витрати), суто фінансові витрати, наприклад інвестиційні платежі, і та-



**Рис. 6.3.** Ієрархія виробничих планів у системі MRP II

кий важливий планово-фінансовий параметр, як графік фінансових потоків. Те, що підлягає аналізу, - це загальний фінансовий результат від основної діяльності за плановий період.

На початку 90-х років у MRP II-системи додали можливості фінансового аналізу, управління конструкторськими розробками, обліку основних засобів, у результаті чого з'явився новий стандарт управління ERP (Enterprise Resource Planing) - набір інтегрованих застосувань, що дозволяє створити інтегроване інформаційне середовище для автоматизації планування, обліку, контролю, аналізу всіх бізнес-процесів підприємства.

#### **6.4. ERP-система (планування ресурсів підприємства)**

Інтегровані рішення за назвою "комплексне планування ресурсів підприємства ERP" покращують якість прийняття рішень, продуктивність і прибутковість. ERP-системи дозволяють ефективно планувати комерційну та виробничу діяльність підприємства. До особливостей застосування сучасних ERP-систем належать:

- автоматизація різноманітних методів планування й управління бізнес-процесів від системи замовлень до масового виробництва з можливістю їх раціонального поєднання та налагодження на специфіку конкретного підприємства;

- інтегроване використання підсистем обліку, аналізу і планування збуту, виробництва, постачання і фінансування;

– реалізація сучасної технології бюджетування та забезпечення динамічного узгодження необхідних ресурсів по всьому спектру бізнес-процесів на основі управлінського обліку витрат і аналізу консолідованої звітності;

– бізнес-планування та управління замовленнями й проектами з урахуванням можливих ризиків внаслідок непередбачених змін у зовнішньому середовищі чи у ресурсних обмеженнях підприємства.

ERP - концепція узгодженого рішення завдань обліку, контролю, планування й управління виробничими і фінансовими ресурсами підприємства. ERP-система - інтегрована інформаційна система управління, що дає змогу створити єдине інформаційне середовище для автоматизації планування, обліку, контролю, управління й аналізу всіх основних господарських процесів підприємства, що реалізовує концепцію ERP.

ERP - термін, введений дослідницькою фірмою Gartner Group для опису систем управління. Ці системи мають забезпечувати автоматизацію процесів планування, прогнозування й управління фінансами, виробництвом, матеріально-технічним постачанням і збутом, бухгалтерським обліком, а також проектування продукції і розробку технологічних процесів тощо. ERP є світовим стандартом управління, запропонованим Американським співтовариством управління виробництвом і запасами.

Відповідно до словника APICS, термін ERP-система використовується у двох значеннях: 1) це інформаційна система для ідентифікації і планування всіх ресурсів підприємства, що необхідні для здійснення продажу, виробництва, закупівель і обліку в процесі виконання клієнтських замовлень; 2) це методологія ефективного планування й управління всіма ресурсами підприємства, які необхідні для здійснення продажу, виробництва, закупівель і обліку при виконанні замовлень клієнтів у сферах виробництва, дистрибуції і надання послуг.

Абревіатура ERP використовується для позначення комплексних систем управління підприємством. Ключовий термін ERP є підприємство, і тільки потім - планування ресурсів. Дійсне призначення ERP - в інтеграції всіх відділів і функцій компанії в єдиний інформаційний простір задяки комп'ютерній мережі, що зможе обслужити всі специфічні потреби окремих підрозділів.

Крім того, для ERP-систем, практично, обов'язковою є наявність можливості електронного обміну даними з іншими застосуваннями, а також моделювання ситуацій, пов'язаних з плануванням і прогнозуванням.

Типова ERP-система включає такі підсистеми:

- виробництво;
- постачання і збут;
- управління запасами;
- післяпродажне обслуговування виробленої продукції;
- кадри;
- наукові дослідження і конструкторські розробки;

– фінанси.

Відповідно до сучасних вимог APICS, ERP-система крім ядра, що реалізовує стандарт MRP II, має включати такі модулі:

- 1) управління логістичними ланцюжками SCM;
- 2) планування і складання виробничих графіків APS (Advanced Planning and Scheduling);
- 3) управління відносинами з клієнтами CRM;
- 4) електронної комерції (Electronic Commerce);
- 5) управління даними про виріб - PDM (Product Data Management);
- 6) платформи Business Intelligence, що включає рішення на основі технологій OLAP і DSS (Decision Support Systems);
- 7) модуль, що відповідає за конфігурацію системи (Stand Alone Configuration Engine, SCE);
- 8) планування ресурсів FRP (Finite Resource Planning). На рис. 6.4 показано основні функціональні модулі, що входять до складу ERP-систем.



Рис. 6.4. Модулі ERP-систем

Основні функції ERP-систем:

- ведення конструкторських і технологічних специфікацій, що визначають склад виробів, а також матеріальні ресурси й операції, необхідні для його виготовлення;
- формування планів продажу і виробництва;
- планування потреб у матеріалах і комплектуючих, термінів і обсягів постачань для виконання плану виробництва продукції;

- управління запасами і закупівлями: ведення договорів, реалізація централізованих закупівель, забезпечення обліку та оптимізації складських і цехових запасів;

- планування виробничих потужностей від укрупненого планування до використання окремих верстатів і устаткування;

- оперативне управління фінансами, включаючи складання фінансового плану і здійснення контролю за його виконанням, фінансовий та управлінський облік;

- управління проектами, включаючи планування етапів і ресурсів, необхідних для їх реалізації.

Основними відмінностями-ERP систем від MRP II-систем можна вважати:

- більшу кількість типів виробництв і видів діяльності підприємств та організацій;

- планування ресурсів за різними напрямками діяльності;

- можливість управління групою автономно працюючих підприємств, корпоративними структурами;

- більше уваги приділено підсистемам фінансового планування й управління;

- наявність функцій управління транснаціональними корпораціями, включаючи підтримку кількох часових поясів, мов, валют, систем бухгалтерського обліку;

- більше уваги приділяється створенню інформаційної інфраструктури підприємства, гнучкості, надійності, сумісності з різними програмними платформами;

- інтегрованість із застосуваннями та іншими системами, що використовуються підприємством, такими як системи автоматизованого проектування, автоматизації управління технологічними процесами, електронного документообігу, електронної комерції;

- наявність у системі або інтеграція з програмними засобами підтримки прийняття рішень;

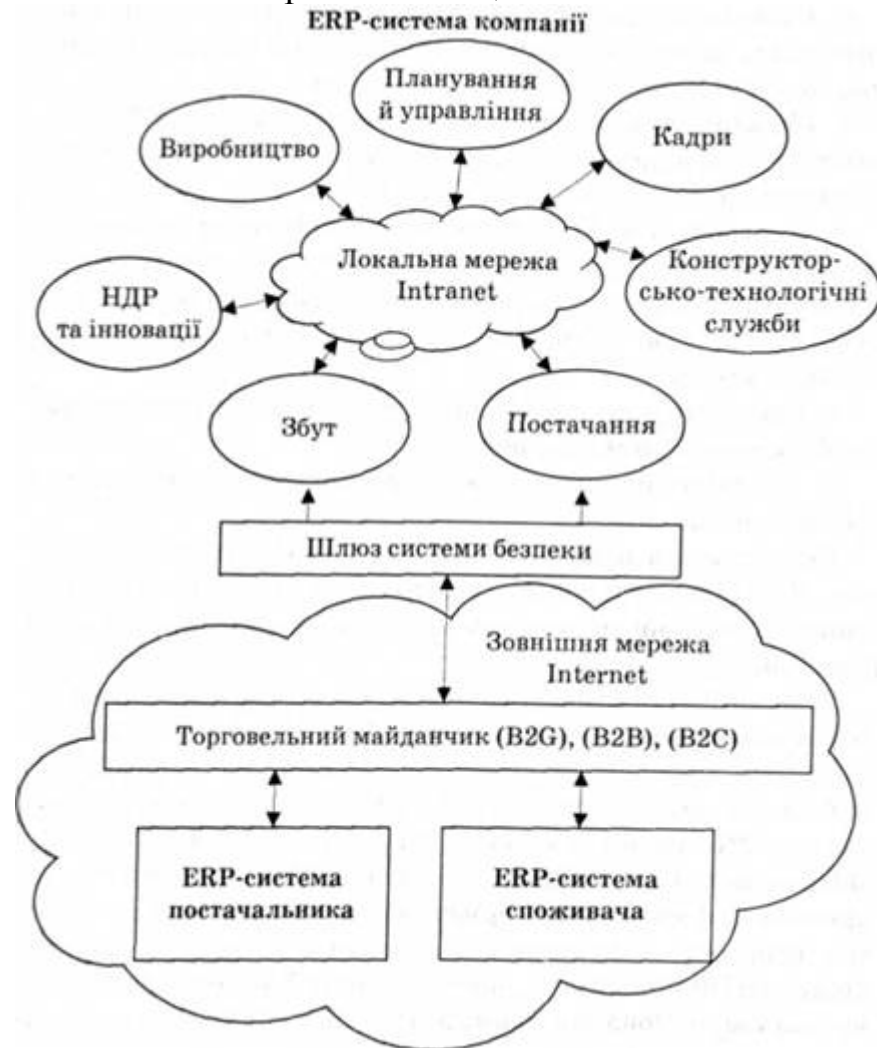
- наявність розвинених засобів конфігурації апаратних і програмних засобів.

Більше уваги надано засобам інтеграції зі сховищами даних. У деяких ERP-системах розроблено засоби інтеграції та адаптації з іншими програмними продуктами і системами (рис. 6.5).

Традиційно ERP-системи будуються на базі тривірневої архітектури клієнт - сервер, коли між сервером бази даних і клієнтами знаходиться сервер застосувань, що відповідає за здійснення всієї бізнес-логіки системи. Процедура впровадження ERP-системи включає створення робочої групи щодо впровадження ІС, виявлення інформаційних потоків на підприємстві, формування нормативно-довідкової бази документів, побудову концептуальної, логічної і фізичної моделей діяльності підприємства, опис

оптимізації бізнес-процесів підприємства, виконання пілотного проекту, тобто етап попереднього впровадження.

На цій стадії повністю моделюється вся діяльність підприємства, в окремих підрозділах у систему вводяться фактичні дані і послідовно тестуються бізнес-функції шляхом моделювання реальних ситуацій, відпрацьовуються взаємні роботи підрозділів на основі тестових прикладів. Протягом адаптації системи на підприємстві здійснюється налагодження системи відповідно до проекту впровадження і тестування окремих модулів у групі впроваджень. Кінцеві користувачі при цьому навчаються працювати з настроєною системою на своїх робочих місцях.



**Рис. 6.5.** Зв'язок різних ERP-систем

У процесі дослідної експлуатації системи перевіряють повністю функціональність усіх модулів і вводять систему у промислову експлуатацію.

В Україні підприємств, які впровадили ERP-системи, невелика кількість, що пов'язано з відсутністю коштів на такі дорогі проекти, неефективністю впровадження (70 % проектів впровадження ERP-систем завершуються невдало), певною складністю ефективної інтеграції ERP-систем з іншими застосуваннями, у тому числі при інтеграції із системами електронного бізнесу (B2B, B2C), обмеженими аналітичними можливостями

ERP-систем і недостатньою підтримкою процесів прийняття рішень, прямих комунікацій між інформаційними системами замовника і постачальника.

Переваги впровадження ERP-систем полягають у зниженні собівартості за рахунок підвищення продуктивності; зменшенні часу виходу товарів і послуг на ринок; скороченні кількості виробничих операцій; зменшенні складських запасів; підвищенні якості продуктів; зростанні інформованості керівництва; підвищенні якості прогнозу і планування; формалізації бізнес-процесів компанії; інтеграції всіх підрозділів компанії.

Можна назвати низку причин, завдяки яким підприємства беруться за впровадження ERP-систем: для інтеграції фінансової інформації та інформації про замовлення, стандартизації і прискорення процесу виробництва, зменшення складських запасів, стандартизації інформації про персонал.

Можна виділити наступні основні причини, внаслідок яких підприємства у всьому світі прагнуть до впровадження систем ERP:

- інтеграція всіх бізнес-процесів підприємства за єдиними правилами і забезпечення оперативного отримання інформації керівництвом про всі сторони діяльності підприємства;

- можливість заміни безлічі успадковуваних автономних корпоративних застосувань, що не задовольняють вимогам сучасного бізнесу, однією ERP-системою.

За оцінками аналітиків, запровадження MRP II/ERP-систем призводить до: скорочення складських запасів підприємства на 8-35 %, зростання продуктивності праці на 8-27, збільшення кількості виконаних замовлень на 7-20 %.

Основні проблеми впровадження ERP-систем полягають у такому: неефективності впровадження, складності ефективної інтеграції ERP-систем із застосуваннями інших фірм, обмеженні аналітичної можливості ERP-систем і недостатній підтримці процесів прийняття рішення, недостатніх можливостях щодо генерації складних інформаційних потоків.

У ERP-системах автоматично планується завантаження виробничих потужностей підприємства відповідно до тих параметрів, які були сформовані на основі наявних замовлень. Безпосередньо в процесі прийому замовлення ERP-система розраховує собівартість і ціну одиниці товару, при цьому враховуються всі компоненти собівартості виробництва і витрати на доставку відповідно до індивідуальних вимог клієнта. Час і витрати, необхідні для переналагодження устаткування при зміні асортименту продукції, що випускається, також враховуються ERP-системою при плануванні виробництва і розрахунку собівартості.

Це дозволяє виробникам переходити на роботу за індивідуальними замовленнями. Відповідно інтегруються автоматизовані системи управління, при цьому саме електронна комерція дозволяє автоматизувати найбільш трудомісткі етапи комерційної діяльності. ERP-системи, що забезпечують ресурсне планування й інтегроване управління всіма бізнес-процесами, доцільно розглядати як основу Internet-рішень компанії. Відсутність надійної

системи внутрішнього планування і контролю, інтегрованої із зовнішнім Internet-рішенням, прирікає компанію на невдачу в новій економіці.

Є кілька шляхів упровадження ERP: власними силами, завдяки адаптації інтегрованих ERP-систем або використання комерційного ПЗ. Відомі ERP-системи різних компаній, зокрема SAP, Oracle, Ахарта тощо, характеристику яких буде подано нижче.

**System21** - система, що повністю відповідає поняттю Enterprise Resource Planning (ERP) і Supply Chain Management (SCM). System21 протягом 20 років працює більше ніж у 2000 компаніях світу і лідирує серед ERP-систем на базі платформи iSeries (AS/400) у секторі середнього бізнесу. Її основні модулі: "Головна книга", "Виробництво", "Логістика", "Фінанси", "Дебітори", "Кредитори", "Матеріальний облік", "Постачання", "Збут", "Планування".

Базисом **iRenaissance** компанії ROSS Systems є концепція управління ресурсами підприємства ERP (Enterprise Resource Planning) й управління ланцюжками постачань SCM (Supply Chain Management). Вони дають змогу раціонально управляти підприємством і звести до мінімуму витрати виробництва.

Використання IC iRenaissance корпорації ROSS Systems допомагає досягти найбільш високих показників при використанні її на середніх і великих підприємствах з процесним типом виробництва, до якого можна віднести такі галузі промисловості: харчову, хімічну, металургійну, нафтопереробну, целюлозно-паперову, фармацевтичну тощо.

Крім того, iRenaissance використовується для автоматизації управління в медичних, адміністративних та освітніх установах, у бібліотеках, у сфері збуту.

Система iRenaissance має такі властивості:

- 1) швидке впровадження - власна методологія FrontRun-ner дає змогу скоротити терміни впровадження системи до 3- 6 місяців;
- 2) відносно низька вартість експлуатації;
- 3) легкість адміністрування системи.

Основні завдання, вирішенню яких сприяє iRenaissance:

- планування обсягів виробництва;
- облік готової продукції;
- зберігання і пошук управлінської інформації (постачальники, матеріали, устаткування);
- розрахунок витрат, калькуляція собівартості продукції;
- організація і планування виробничого процесу;
- управління фінансами і контроль взаєморозрахунків з постачальниками і споживачами;
- бухгалтерський облік на підприємстві і в його підрозділах;
- аналіз і прогноз продажу;
- контроль за використанням кредитів;
- планування й аналіз бюджету підприємства та його підрозділів;



- розрахунок, аналіз, узагальнення і прогноз фінансових показників підприємства;
- розрахунок грошових потоків на основі бухгалтерських звітів;
- управління перевезеннями та оптимізація транспортних витрат;
- управління експлуатацією устаткування;
- організація електронного документообігу;
- ведення електронного архіву підприємства;
- облік кадрів, управління персоналом;
- облік трудових витрат і розрахунок заробітної плати;
- управління проектами.

Система складається з: виробничого, фінансового модулів, модулів управління персоналом і розрахунку зарплати тощо. Ці модулі можуть використовуватися незалежно один від одного для автоматизації окремих напрямів діяльності підприємства. Впровадження і Renaissance дає змогу оптимізувати роботу підприємства - позбавитися непотрібних і дубльованих операцій, запасів, тобто скоротити витрати і збільшити прибуток.

Систему іRenaissance легко налагодити відповідно до офіційних вимог щодо документообігу і звітності різних країн. Вона має вбудований репозиторій знань (бізнес-процедур), в арсеналі якого зберігаються різні бізнес-процедур. іRenaissance повністю локалізована, що підтверджується успішно завершеними проектами з упровадження, й успішною експлуатацією системи на російських підприємствах.

Система підтримує кишенькові ПК і пристрої для зчитування штрихкодів. Клієнт-серверна архітектура іRenaissance повністю відповідає таким, що пред'являється до вимог КІС з надійності і безпеки. Крім того, вона є мас штабова пою, тобто їй властиве поступове нарощування конфігурації у міру підключення нових, призначених для користувача, робочих місць. Вся корпоративна інформація зберігається в єдиній базі даних системи, що дає змогу організувати сховище даних та аналізувати накопичену інформацію за допомогою засобів підтримки прийняття рішень, наприклад за допомогою DSSuite, власної системи ROSS Systems.

іRenaissance є відкритою системою. Це означає, що користувач може додавати до неї необхідні функції, використовуючи вбудовані CASE-засоби і мову четвертого покоління (4GL).

#### **Основні модулі іRenaissance:**

- o іRenaissance Connect - засіб швидкої розробки Web-інструментарію, що забезпечує зв'язок з ERP-платформою підприємства;
- o іRenaissance Financial - управління фінансами компанії;
- o іRenaissance Distribution - управління продажем;
- o іRenaissance Manufacturing - управління виробництвом;
- o іRenaissance Maintenance - управління експлуатацією майна;
- o іRenaissance Human Resources/Payroll - управління персоналом і розрахунок заробітної плати;
- o іRenaissance CS Materials Management - управління матеріалами;

o iRenaissance Transportation Manager - управління перевезеннями і транспортними витратами;

o iRenaissance DSSuite - засіб підтримки прийняття рішень;

o iRenaissance SAM - засіб моделювання й опису бізнес-процесів.

Система iRenaissance підтримує платформи Windows, Unix, OPENVMS, СУБД Oracle, MS SQL Server, Sybase SQL Server, RDB.

**Microsoft Dynamics AX** - комплексне ERP-рішення, створене спеціально для середніх і великих компаній, яке допомагає розширити свої можливості та отримати нові конкурентні переваги.

**Microsoft Axapta** має такі переваги: це система, що дає змогу вести бізнес саме так, як клієнт вважає за потрібне; покращує взаємодію з клієнтами, діловими партнерами і співробітниками; надає потужну, вичерпну функціональність в єдиній інтегрованій системі; надає можливості для швидкого зростання і розвитку бізнесу.

Microsoft Axapta охоплює всі сфери бізнесу, включаючи виробництво і дистрибуцію, управління ланцюжками постачань і проектами, фінансовий менеджмент і засоби бізнес-аналізу, управління відносинами з клієнтами й управління персоналом.

Система відповідає всім вимогам міжнародних стандартів бухгалтерського обліку і законодавства, може працювати з багатьма мовами і з різними валютами.

В основі Microsoft Business Solutions-Axapta лежать найсучасніші новітні технології управління і високотехнологічні рішення, що дозволяють ефективно управляти підприємством. Система підходить для автоматизації бізнес-процесів у межах управлінського обліку переважно для середніх і великих підприємств різних сфер господарської діяльності.

Axapta - це ERP-система, що працює в середовищі електронного бізнесу. Унікальність системи Axapta полягає в тому, що її сучасна технологія забезпечує єдиний інформаційний простір підприємства, в якому бек-офіс і фронт-офіс працюють як єдине ціле. Axapta пропонує набір можливостей для бізнес-аналізу, що удосконалює процес прийняття рішень і комплексне управління відносинами з клієнтами (CRM).

Основними модулями системи Axapta є фінанси; торгівля і логістика; виробництво; електронна комерція; управління персоналом; проекти; управління відносинами з клієнтами; управління знаннями; управління логістичними ланцюжками.

Великий набір функціональних можливостей системи Axapta дозволяє одержати низку певних переваг: менші витрати на створення і підтримку системи; легкість в оновленні за-стосунків; баланс надмірної інформації; повна інтеграція бізнес-процесів.

**Microsoft XAL** - це інтегрована автоматизована система з універсальним набором функцій для управління фінансово-господарською діяльністю підприємств середнього і великого бізнесу. Система охоплює практично всі функціональні ділянки підприємства і складається з таких модулів: головна книга, дебітори, кредитори, продаж, закупівля, склад,

основні засоби, зарплата, управління кадрами, проекти і планування/управління виробництвом.

Крім стандартних модулів, є різні доповнення, які розширюють можливості наявних модулів системи. У системі є розвинені засоби планування, аналізу діяльності та оперативного управління підприємством.

Microsoft XAL - багатовалютна система з можливістю отримання управлінської, фінансової і податкової звітності та ведення обліку за міжнародними стандартами GAAP, IAS.

Продукт базується на клієнт-серверних технологіях, у нього вбудовано апаратно-незалежну мову четвертого покоління XAL (extended Application Language), що забезпечує повне масштабування і можливість розвитку системи відповідно до вимог бізнесу, що змінюються. Гнучкість, налагоджуваність і масштабування системи забезпечуються можливістю маніпулювання великою кількістю параметрів, а також наявністю вбудованих інструментів налагоджування.

Механізм відстежування призначених для користувача змін зберігає початкові версії змінних елементів і забезпечує безпеку для розробників, оскільки завжди можна відновити функціональність, що існувала в початковій системі.

При зіставленні з Microsoft XAL/ERP-систем, близьких за функціональністю тиражованих програмних рішень для управління бізнесом, таких як SunSystems, Platinum, Exact, Microsoft XAL відрізняється мінімальними інвестиціями, необхідними для установки і супроводу системи.

Оскільки система Microsoft XAL побудована відповідно до правил ведення бізнесу на західних підприємствах, для підприємств вона може слугувати не тільки засобом автоматизації, а й засобом передачі західних стандартів ведення бізнесу.

Конструкція Microsoft XAL орієнтована на те, щоб продукт можна було легко і швидко доопрацювати з урахуванням галузевих та індивідуальних вимог конкретного замовника.

У Microsoft XAL реалізовано практично всі основні методи, за допомогою яких можна розраховувати обсяги виробництва на основі одержаних замовлень і плану реалізації, визначати параметри завантаження виробничих потужностей, приймати рішення щодо використання наявних ресурсів тощо.

#### **6.4.1. Застосування системи Microsoft XAL на світових ринках**

Спектр ділової активності компаній, що зробили свій вибір на користь автоматизованої системи Microsoft XAL охоплює практично всі види виробничої діяльності, починаючи з важкого машинобудування і закінчуючи виробництвом високоточного вимірювального устаткування, будівництвом, виробництвом фармацевтичних продуктів, виробництвом товарів народного споживання і побутової техніки.

Microsoft XAL використовується також у сфері послуг, зокрема консалтингу, телекомунікаційному і готельному бізнесі, туризмі, візовій підтримці тощо. За допомогою Microsoft XAL управляють своєю діяльністю

підприємства добувних та оброблювальних галузей, таких як нафтовидобування і нафтопереробка, лісова (деревообробна) промисловість, газодобування.

У загальному обсязі користувачів Microsoft XAL у країнах Західної Європи торгові підприємства становлять 40 %, промислові - 34, підприємства сфери послуг - 17, державні підприємства - 9 %.

**Oracle E-Business Suite** - це повний комплекс застосувань для електронного бізнесу, що функціонує у межах локальної мережі Intranet і глобальній мережі Internet. Комплекс включає повний набір рішень, необхідних для автоматизації управління підприємством.

Oracle E-Business Suite дає змогу на єдиній платформі вирішувати широкий спектр задач: управління ефективністю підприємства на основі системи корпоративних показників; бюджетування і консолідація; облік і звітність; управління виробництвом, запасами, ланцюжками постачань, персоналом, якістю, продажем.

Пакет бізнес-застосувань Oracle E-Business Suite включає сотні інтегрованих програмних модулів, що дозволяють підприємству вирішувати бізнес-задачі у сфері управління виробництвом, фінансами, матеріально-технічним постачанням, запасами і збутом, маркетингом і продажем, взаємодією з постачальниками і відносинами з покупцями, а також ефективно будувати кадрову політику, управлінський облік і проводити операції через електронні торговельні майданчики. Функціональні блоки Oracle E-Business Suite 1 li:

- о Oracle ERP (Enterprise Resource Planning) - застосування для автоматизації управління внутрішньогосподарськими процесами підприємства (виробництво, фінанси, постачання, управління персоналом) та їх оптимізації;

- о Oracle CRM (Customer Relationship Management) - застосування для автоматизації і підвищення ефективності процесів, орієнтованих на відносини з клієнтами (продаж, маркетинг, послуги);

- о Oracle E-Hub (Електронна комерція) - застосування організації електронних торговельних майданчиків.

Застосування Oracle CRM, Oracle ERP, Oracle E-Hub (Exchange) повністю інтегровані і створені для співпраці один з одним, утворюючи повний єдиний комплекс для електронного бізнесу - Oracle E-Business Suite, що дозволяє підприємствам використовувати єдине джерело даних у системі та не витратити час і кошти на інтеграцію.

Основні можливості функціональних блоків: Oracle ERP; управління дискретним виробництвом та безперервним виробництвом, фінансами, персоналом, постачанням і складами, проектами; Oracle CRM; маркетинг, продажі сервіс, центр взаємодії (Call-center), електронні торговельні майданчики (Exchange).

Корпорація Oracle є постачальником корпоративного програмного забезпечення. Вона пропонує повний комплекс технологій для побудови IT-інфраструктури й управління сучасним підприємством, зокрема сімейство

базових програмних технологій Oracle 10g, готове рішення для колективної роботи Oracle Collaboration Suite, повнофункціональний комплекс бізнес-застосувань Oracle E-Business Suite та інтеграційне рішення для управління даними Oracle Data Hub. Корпорація надає свої продукти і послуги у сфері консалтингу, навчання і технічної підтримки у багатьох країнах світу.

*my SAP Business Suite* - це сукупність рішень, що пропонує широку функціональність, повну інтеграцію, необмежену масштабованість і взаємодію в рамках мережевих інфраструктур ведення бізнесу.

Сфера застосування: повнофункціональна ERP-система для комплексної автоматизації середніх і великих компаній. Підтримує управлінські концепції: ERP, EAM, MRP II, CRM, SCM, PLM, eBusiness.

mySAP Business Suite включає: управління ресурсами підприємства (mySAP ERP); управління відносинами з клієнтами (mySAP Customer Relationship Management); управління життєвим циклом продукту (mySAP Product Lifecycle Management); управління відносинами з постачальниками (mySAP Supplier Relationship Management).

Функціональні можливості: управління виробництвом, постачання, управління запасами, взаємодія з постачальниками, управління персоналом і розрахунки заробітної платні, фінансове планування, управлінський облік, електронний бізнес тощо.

Система управління ресурсами підприємства *my SAP Enterprise Resource Planning (mySAP ERP)* - повнофункціональне ERP-рішення, що є набором пакетів і модулів. Це дає змогу компаніям впроваджувати тільки ту функціональність управління бізнесом, яка їм необхідна, що спрощує процес переходу на нові версії і скорочує витрати. Крім того, компанії можуть включати в рішення додаткові функціональні можливості, такі як діяльність на базі порталу, мобільний сервіс, *біз-нес-аналітика* тощо. Можливість підключення нових модулів реалізована за рахунок технології інтеграції SAP NetWeaver.

Рішення mySAP ERP включає mySAP ERP Financials, mySAP ERP Human Capital Management, mySAP ERP Operations, mySAP ERP Corporate Services, mySAP Customer Relationship Management (mySAP CRM).

Рішення для планування, формування і підтримки взаємовигідних зв'язків з клієнтами дає змогу пов'язувати персонал, бізнес-процеси та інформацію про клієнтів у єдиному інформаційному просторі.

Функціональність mySAP CRM Interaction Center дозволяє взаємодіяти з клієнтом незалежно від типу контакту - телефоном, факсом, електронною поштою або через Internet. Можна також використовувати рішення mySAP CRM з мобільними засобами зв'язку, такими як портативний комп'ютер, мобільні телефони і PDA. Рішення забезпечує цілісність інформації в будь-якому розрізі даних.

Крім того, рішення mySAP CRM надає персоналу можливість мати різнорівневий доступ до інформації про ринок, аналітику, застосування.

Оперативні можливості CRM дають змогу управляти і синхронізувати всі контакти з клієнтами у сфері маркетингу, продажу і послуг. Аналітичний

CRM допомагає оптимізувати всі інформаційні джерела для кращого розуміння поведінки клієнта. Таким чином, mySAP CRM дозволяє взаємодіяти і співпрацювати з постачальниками, партнерами і клієнтами для оптимізації процесів і задоволення потреб клієнта.

Рішення надає повний набір інструментів для задоволення потреб клієнта у будь-який момент циклу взаємодії з клієнтом: від появи нового клієнта в базі даних і багатоканального продажу до процесу замовлення продукту і додаткових послуг клієнтові.

**my S AP Supply Chain Management** (mySAP SCM) - це застосування для управління логістичними ланцюжками (від планування ланцюжків до об'єднання їх у мережу), відкрите, інтегроване рішення, розроблене для спільного ведення електронного бізнесу. Рішення mySAP SCM перетворює управління логістичними ланцюжками з лінійного послідовного процесу в колективну роботу персоналу компанії, партнерів і постачальників на електронному торговельному майданчику, що допомагає їм координувати свої дії у сфері логістики.

Інтеграція mySAP SCM з продуктами SAP і сторонніми продуктами, як всередині, так і за межами підприємства, дозволяє одержувати вигоду від використання платформи mySAP Business Suite для ведення електронного бізнесу. mySAP SCM забезпечує прозорість усіх рівнів запасів, замовлень, прогнозів, виробничих планів та основних показників ефективності незалежно від місцезнаходження постачальника і клієнта.

Інтеграція з функціональністю mySAP Customer Relationship Management надає можливість одержати прозорі ланцюжки постачань за вимогою. Інтеграція з функціональністю mySAP Product Lifecycle Management дозволяє залучати постачальників до процесу логістики з метою оперативного постачання нових Продуктів на ринок. Інтеграція з функціональністю mySAP Business Intelligence допомагає відстежувати процес ланцюжка постачань у режимі реального часу.

**mySAP Product Lifecycle Management** (mySAP PLM) дозволяє управляти життєвим циклом продукту: здійснювати процеси проектування і конструювання, розвитку і вдосконалення продукту, управління проектами, активами серед біз-нес-партнерів. Рішення mySAP PLM інтегрує процеси діяльності всіх учасників розробки продукту: проектувальників, постачальників, виробників і клієнтів. Для всього життєвого циклу продукту, що включає розробку первинної концепції продукту, дизайн, проектування, розгортання виробництва, управління змінами продукту, сервіс і супровід, mySAP PLM забезпечує єдиний підхід до бізнес-процесів і надає необхідну інформацію, пов'язану з продуктом. Завдяки цьому mySAP PLM надає компаніям можливість одержати продукт, що відповідає потребам ринку в потрібний час і при раціональних витратах.

Рішення mySAP PLM поставляється разом з рішенням mySAP Enterprise Portals (портал для ролевого використання), для того, щоб персонал мав зручний доступ, що базується на Internet-технологіях, до внутрішньої і

зовнішньої інформації, застосувань і сервісу, що необхідні для виконання їх обов'язків.

Інтеграція рішення mySAP PLM з mySAP Supply Chain Management, mySAP Customer Relationship Management, mySAP E-Procurement і mySAP Marketplace надає можливість здійснювати спільне проектування і конструювання, дизайн і закупівлі.

**my S AP Supplier Relationship Management** - рішення з управління відносинами з власними постачальниками, що дозволяє зменшити вартість товарів, оскільки допомагає у виборі оптимальних джерел постачання і забезпечує швидке повернення інвестицій. Рішення дозволяє інтегрувати процеси, розширювати співпрацю й автоматизувати взаємодію з кожним із постачальників.

mySAP SRM призначено для автоматизації розширеного циклу бізнес-процесів, починаючи від вироблення стратегії постачання до оперативних закупівель і залучення постачальників. Рішення базується на консолідованому й уніфікованому бізнес-контенті, надає можливість тіснішої співпраці з усіма діловими партнерами, дозволяє оптимізувати оперативні процеси пошуку і вибору джерел постачання, скорочувати час закупівельного циклу й одержувати вигоду від служб матеріально-технічного постачання. mySAP SRM інтегрується з іншими модулями mySAP Business Suite. Підтримується обмін даними і документами з ERP- і SCM-системами (як виробництва SAP, так і інших розробників).

**mySAP Human Resources** - рішення для управління персоналом mySAP HR синхронізує й оптимізує всі бізнес-процеси компанії відповідно до вимог законодавства і бізнесу. Рішення mySAP HR дає змогу відділу управління персоналом тісніше й ефективніше співпрацювати з виробничими менеджерами. Рішення об'єднує стратегічні цілі управління персоналом з аналітичними даними ринку праці і допомагає об'єктивно оцінити інвестиції в персонал.

Рішення mySAP HR надає можливість: залучати, утримувати і мотивувати досвідчений персонал; домагатися того, щоб індивідуальні цілі співробітників збігалися з корпоративною стратегією розвитку підприємства; організувати систему оплати праці, що відповідає вимогам ринку і внутрішньої політики; виробляти інші стратегії і рішення управління персоналом, які підвищують продуктивність компанії.

**my SAP Financials** - закінчене фінансове рішення, яке дозволяє: проводити довгострокове управління у вартісному відношенні; інтегрувати всю фінансову інформацію та інформацію з продуктивності, процеси управління для успішного прийняття стратегічних рішень; поліпшити спілкування з інвесторами; упорядкувати фінансові потоки в ланцюжку постачань; взаємодіяти з клієнтами і постачальниками з питань платежів; зменшувати операційні витрати; збільшувати ефективність виробництва.

**mySAP Business Intelligence** (mySAP BI) - комплексне програмне рішення у сфері бізнес-аналітики, що забезпечує швидкий доступ до інформації і використання її в прийнятті стратегічно важливих рішень.

Рішення mySAP BI пропонує можливість управляти знаннями, які допомагають компанії об'єднувати "тих, хто знає" з "тими, кому потрібно дізнатися". Основною компонентою його є сховище даних, розроблене для зберігання внутрішньої і зовнішньої інформації, що включає документацію, відео- й аудіо- кліпи. Рішення mySAP Business Intelligence об'єднує інформацію по всій платформі mySAP Business Suite. Воно також надає можливість швидко реагувати на зміни ринку, контролювати показники основних чинників успіху, аналізувати й оптимізувати продуктивність підприємства на основі єдиної бізнес-моделі.

Рішення mySAP BI включає такі аналітичні застосування:

- o SAP Business Information Warehouse (SAP BW) - сховище бізнес-інформації;

- o SAP Knowledge Management (SAP KM) - рішення для управління знаннями;

- o SAP Strategie Enterprise Management (SAP SEM) - рішення для управління корпоративною стратегією.

Рішення *mySAP Enterprise Portal* надає можливість об'єднання всієї актуальної на певну дату інформації і надання її користувачам. У результаті персонал компанії може використовувати весь цей обсяг інформації для прийняття більш обґрунтованих бізнес-рішень, здійснювати успішну співпрацю і збільшувати продуктивність та ефективність виробництва.

Рішення mySAP Enterprise Portal дає змогу:

- o управляти всіма видами корпоративних даних, включаючи дані бізнес-застосувань, корпоративні бази даних, архів документів та інформацію з Internet;

- o консолідувати всю корпоративну інформацію, незалежно від її місцезнаходження;

- o надавати користувачам повний зміст інформації, використовуючи Push-технологію, відповідно до їх конкретних функціональних обов'язків та інтересів;

- o одержувати аналітичні дані і використовувати їх для досягнення стратегічної мети;

- o працювати спільно і виконувати персональну роботу;

- o мати доступ до всіх застосувань з єдиним загальним паролем.

*mySAP Mobile Business* - це закінчене рішення для ведення мобільного бізнесу. mySAP Mobile Business надає рішення для всіх сучасних засобів мобільного зв'язку як у режимі он-лайн, такі в режимі оф-лайн. Основні компоненти включають: широкий спектр застосувань, розроблених для ведення мобільного бізнесу; відкриту технологію стандартів галузевих рішень.

Міжнародна компанія **SAP AG** (заснована в 1972 р., штаб-квартира розташована у Вальдорфе, Німеччина) є найбільшим у світі розробником міжкорпоративного програмного забезпечення (детальніше див. у розд. 10).

У SAP працює понад 30 тис. чоловік у 50 країнах світу. Компанія SAP є визнаним лідером серед постачальників рішень для сумісного ведення



бізнесу, призначених для всіх типів промисловості. Компанія представлена на кількох біржах, включаючи Франкфуртську і Нью-йоркську фондові біржі, під символом "SAP".

У СНД і країнах Балтії фірма SAP AG працює з 1992 р. Компанія має адаптований російськомовний продукт і великий досвід впроваджень.

Користувачі системи: міністерства фінансів і оборони багатьох країн, армія і ВМФ США, NASA, компанії Reebok, Uhde, Colgate.

Основний продукт компанії,- ERP-система SAPR/3, в якому реалізовано більше, ніж 1000 бізнес-процесів. На сьогодні розроблено більше 50 версій цієї системи на 28 мовах. За оцінкою Gartner Group, система SAP R/3 призначена для великих компаній з річним оборотом від 200 млн дол. США (детальніше див. у розд. 10.4).

R/3 впроваджено в різних галузях: аерокосмічній і оборонній; автомобілебудуванні; банківській справі; хімічній промисловості; виробництві споживчих товарів; проектуванні і будівництві; охороні здоров'я; страхуванні; ЗМІ; фармацевтиці; роздрібній торгівлі тощо.

Наприклад, SAP R/3 встановлено в таких компаніях: Autogrill SPA, Chevron, Colgate, Palmolive, COMPUSA, Deutsche Telekom AG, Eastman Chemical, ENI SPA, Fiat SPA, Microsoft, Minolta, Mott's, Pirelli SPA, Robert Bosch GMBH, Royal Philips Electronics, Security National Servicing, Siemens AG, Sony, Statoil, Telecom Italie SPA і Volkswagen AG.

У 90-х роках компанія SAP AG почала приділяти велику увагу Internet-орієнтованим рішенням своїх програмних продуктів, удосконалюючи R/3 за рахунок доповнення її клієнтськими Internet-орієнтованими модулями, поліпшуючими функціонування системи в гетерогенних середовищах.

Головний акцент SAP AG - Internet-портал MYSAP.com, за допомогою якого за запитами клієнтів надається відкрите середовище персональних рішень для спільного ведення електронного бізнесу і ПЗ для онлайн-діяльності.

На mySAP.com можна як купувати, так і продавати, а представники груп за інтересами можуть знаходити одне одного з метою торгівлі продуктами і послугами. Сервіси доступу до застосувань користувачів надаються через "тонкого" клієнта.

Послуги mySAP.com (CRM, SCM) можна одержувати також через міні-комп'ютери і мобільні телефони. Крім того, пропонується хостинг інших систем на основі XML, WML і HTML-технологій. Архітектурно mySAP.com є програмно-апаратним ядром, незалежна системна надбудова якого написана мовою ABAP, що підтримує бізнес-логіку і робочий простір, який включає засоби адміністрування, безпеки тощо. Така структура дає змогу збирати систему з готових логічних блоків і розширювати її можливості, не піклуючись про цільову платформу.

**SAP Business One** - це готове програмне рішення для бізнесу. SAP Business One - перспективна пропозиція в лінійці рішень SAP. Це самостійне рішення, призначене для підприємств середнього і малого бізнесу, що працюють у сферах торгівлі й надання сервісних послуг. Передусім це

управлінська система, що дозволяє автоматизувати роботу відділу продажу (SFA), закупівлі, ведення складського обліку і фінансів.

Основні можливості SAP Business One: фінанси, план рахунків, сегментація рахунків, підтримка бухгалтерських операцій, періодичні проводки, бухгалтерські звіти, звіти підприємства, порівняльні звіти, бюджет, облік витрат, облік курсових різниць, шаблони фінансових звітів, можливості продажу, аналіз можливостей продажу, прогноз продажу, управління контактами, пропозиція, замовлення, відвантаження, повернення, продаж, рахунок-фактура, попередня фактура, розрахунок прибутків за документом, аналіз продажу, закупівлі, надходження, додаткові витрати, розрахунки, вхідні платежі, платіжна система, виписки банку і перевірка, запаси, довідник товарів і послуг, управління запасами, операції зі складу, прайс-лист, спеціальні знижки, спеціальні ціни, відбір і упаковка, відомість комплектування, складські звіти, специфікації, замовлення-наряд, база рішень, облік витрат, персонал, основні дані співробітників, список співробітників тощо.

Переваги для бізнесу: підвищення доходу за рахунок підвищення продуктивності роботи співробітників, швидке прийняття якісних рішень, гарантована можливість масштабування, глобальний підхід, контроль витрат, управління можливостями продажу, мінімальні витрати на впровадження. Термін впровадження - від кількох днів до двох місяців.

Відмітні особливості SAP Business One від багатьох інших пропозицій на ринку середнього і малого бізнесу:

- о адаптація під конкретне підприємство здійснюється не шляхом програмування, а налагодження перевірених бізнес-процесів. Це сприяє скороченню термінів впровадження рішення до 6-10 тижнів, включаючи передпроектне обстеження і навчання користувачів;

- о замовникові поставляється єдине за функціональністю рішення, а не набір модулів з нечіткою перспективою їх інтеграції між собою;

- о у рішенні - своя власна вбудована функціональність CRM;

- о генератор звітів дозволяє будувати аналітичні звіти без допомоги програмування;

- о для розширення функціональності і/або інтеграції зі спеціалізованими рішеннями існує спеціальний набір інтерфейсів програмування (Software Development Kit);

- о може бути налагоджена автоматична перевірка виконання бізнес-правил, або правил документообігу.

Інтеграція може бути здійснена як на рівні даних (для звітності), так і на рівні бізнес-процесів (наприклад, центральна бухгалтерія, замовлення на склад постачальника).

**ІС:Підприємство 8.0** включає платформу і прикладні рішення для автоматизації діяльності підприємств та приватних осіб.

Сама платформа не є програмним продуктом для використання кінцевими користувачами, які зазвичай працюють з одним з багатьох прикладних рішень (конфігурацій), розроблених на даній платформі. Такий

підхід дає змогу автоматизувати різні види діяльності підприємств, використовуючи єдину технологічну платформу.

Склад прикладних механізмів "1С:Предприятие" орієнтований на вирішення задач автоматизації обліку й управління підприємством. Використання проблемно орієнтованих об'єктів дозволяє розробникові вирішувати задачі складського, бухгалтерського, управлінського обліку, розрахунку зарплати, аналізу даних та управління на рівні бізнес-процесів.

Основними завданнями при розробці платформи "1С:Предприятие 8.0" є:

- о забезпечення високої ефективності використання системи для широкого кола підприємств за рахунок розширення функціональності, підвищення продуктивності і масштабування;

- о збереження початкового рівня масових рішень від малих підприємств й індивідуальних користувачів до середніх підприємств.

Гнучкість платформи дає змогу застосовувати "1С:Предприятие 8.0" в найрізноманітніших сферах, а саме:

- о автоматизації виробничих і торгових підприємств, бюджетних і фінансових організацій, підприємств сфери обслуговування;

- о підтримки оперативного управління підприємством;

- о автоматизації організаційної і господарської діяльності;

- о веденні бухгалтерського обліку з кількома планами рахунків, регламентованої звітності;

- о можливості для управлінського обліку і побудови аналітичної звітності, підтримки багатовалютного обліку;

- о рішенні задач планування, бюджетування і фінансового аналізу;

- о розрахунку зарплати й управління персоналом;

- о інших сферах.

"1С:Предприятие 8.0" має широкі можливості взаємодії з іншими застосуваннями і побудови складних інтегрованих рішень, серед яких можна назвати такі:

- о взаємодія через СОМ-з'єднання;

- о підтримка механізму Automation;

- о підтримка Internet-протоколів HTTP, HTTPS, FTP;

- о відправка і прийом електронної пошти (e-mail);

- о робота з XML-документами;

- о обмін даними через текстові файли; зчитування і запис текстових файлів великого розміру;

- о підтримка DBF-файлів;

- о технологія зовнішніх компонент, підтримка роботи з торговим устаткуванням.

Фірма "1С" випускає тиражовані прикладні рішення, призначені для автоматизації типових завдань обліку й управління в комерційних підприємствах реального сектору і бюджетних організаціях. У кожному програмному продукті поєднується використання стандартних рішень і

максимальне урахування специфіки завдання конкретної галузі або виду економічної діяльності підприємства.

Галузеві і регіональні прикладні рішення створюються зусиллями партнерів-розробників і призначені для автоматизації окремих напрямів або сфер діяльності підприємств. Всі вони сертифіковані на вимоги <sup>4</sup>1С:Совместимо".

Тиражовані рішення "1С:Предприятия 8.0" розроблено з урахуванням міжнародних стандартів MRP, MRP II, CRM, SCM, ERP, ERP II.

Корпоративна інформаційна система SIKE це рішення, що дає змогу ефективно управляти бізнесом. Впродовж багатьох років SIKE забезпечує оптимальне використання ресурсів підприємств і досконалі рішення з управління російських підприємств.

**Гнучкість системи SIKE** забезпечується використанням конструктора інформаційних систем SIKE Builder, сумісного із засобами моделювання бізнес-процесів підприємства. Це надає можливість проектування системи будь-якому фахівцеві підприємства.

Супровід і підтримка досягається за рахунок форм підтримки користувачів, які дозволяють закрити всі потреби користувачів корпоративної інформаційної системи SIKE ERP і методології впровадження та супроводу KIC SIKE ERP.

Захист інформації від втрат ґрунтується на системі резервування бази даних і на зберіганні всіх модифікацій записів критично важливих таблиць. У випадку, якщо користувач видаляє доступну для нього інформацію, фізичного видалення записів у базі даних не відбувається, що завжди дає змогу провести аналіз і відновлення.

Надійність системи SIKE гарантується більше ніж 1250 блокуваннями і процедурами самоконтролю, що існують в інформаційній системі.

Принцип одноразового введення не дозволяє різним підрозділам підприємства вести різний облік одного і того самого процесу.

Захист від несанкціонованого доступу є дворівневим захистом інформації: засобами бази даних і програмними засобами системи SIKE. Інформація, що представляє комерційну цінність, зберігається в базі даних у закодованому вигляді.

Інтеграція з іншими системами, що діють на підприємстві, підтримується завдяки гнучкості системи і засобам системи SIKE, що дають змогу виконати якісну "стиківку", практично з будь-яким програмним забезпеченням.

**Галактика Business Suite** - повнофункціональний комплекс бізнес-рішень, що дозволяє в єдиному інформаційному просторі виконувати типові і спеціалізовані завдання управління підприємством, холдингом, групою компаній в умовах сучасної економіки.

Комплекс Галактика Business Suite на основі новітніх інформаційних технологій забезпечує рішення:

о всьому спектра управлінських завдань підприємства відповідно до концепції ERP;

о завдань підтримки прийняття управлінських рішень на базі визначення, планування, досягнення й аналізу ключових показників діяльності підприємства;

о завдань корпоративного управління, пов'язаних із консолідацією інформації, що поступає з територіально-розподілених підрозділів/підприємств, та управлінням розподіленими ресурсами (матеріальними, фінансовими, трудовими).

Для підприємства дуже важлива наявність єдиної інформаційної системи, яка забезпечує ефективне рішення управлінських задач продуктивну і комфортну роботу кожного співробітника, підтримує досягнення бізнес-цілей і дає змогу гнучко реагувати на зовнішні та внутрішні зміни в діяльності підприємства.

Ці властивості повною мірою характерні для комплексу Галактика Business Suite, в якому органічно відображені передові методології і концепції управління" новітні розробки у сфері інформаційних технологій, таких як:

о управління ефективністю бізнесу (Corporate Performance Management, CPM) - рішення задач розподілених багаторівневих бізнес-груп, холдингів і корпорацій. Можливість організації єдиного інформаційного простору не тільки усередині корпорації, але і стосовно клієнтів та партнерів, інтеграції зовнішніх застосунків синхронізації даних і бізнес-процесів;

о підприємство реального часу (Real-Time Enterprise, RTE) - отримання оперативної і достовірної інформації про виконання важливих критичних процесів, допомога у прийнятті рішень у потрібний момент, здатність швидко адаптуватися до потреб бізнес-моделі управління підприємством;

о сервісна шина підприємства (Enterprise Service Bus, ESB) - інтеграція в єдиний інформаційний простір розподілених структур, що функціонують у гетерогенних системах;

о система управління бізнес-процесами (Business Process Management, BPM) - підвищення ефективності функціонування процесів та якості управління, контроль і управління регламентними операціями, динамічне вдосконалення процесів, гнучкість і зручність у роботі;

о мобільний бізнес - побудова мобільного бізнесу з динамічною структурою, інтеграція з партнерами через Internet (організація таких рішень, як B2C і B2B), підтримка мобільних пристроїв, Internet-магазинів.

Ядром комплексу Галактика Business Suite є система Галактика ERP. Можливості системи доповнюються і розширюються за допомогою рішень "Галактики" на платформі Галактика Rnnet, що є готовими продуктами і можуть бути легко адаптовані під потреби конкретного замовника: Галактика Business Intelligence і Галактика Financial Management.

У 2005 р. корпорація "Галактика" випустила на ринок нову версію системи Галактика ERP, яка, зберігаючи переваги попередніх розробок, надає замовникові розширені можливості.

Спеціально для середніх підприємств на базі системи Галактика ERP було розроблено рішення Галактика-Старт, що дає змогу швидко і з

мінімальними витратами провести автоматизацію основних бізнес-процесів підприємства. Невеликі компанії, що динамічно розвиваються, можуть скористатися і спеціальною пропозицією корпорації "Галактика" - рішенням Галактика-прогрес.

Система *Галактика Enterprise Resource Planning* (ERP) - основа комплексу Галактика Business Suite.

Можливості системи дозволяють в єдиному інформаційному просторі оперативно вирішувати головні управлінські задачі, забезпечити менеджерів різного рівня управління необхідною і достовірною інформацією для управлінських рішень, таких як побудова системи обліку і формування різних видів звітності, управління матеріальними і фінансовими потоками, фінансове планування й оперативний фінансовий менеджмент, виробниче планування й управління, контролінг, управління якістю продукції, управління персоналом і реалізація кадрової політики, управління відносинами з клієнтами.

Рішення корпорації "Галактика" вирізняються такими особливостями: відповідність концепції ERP і стандарту MRP II; розвинений функціонал системи слугує гарантією того, що всі обліково-управлінські завдання замовників будуть успішно вирішені; сучасність методик управління й урахування національної специфіки.

Поєднання в системі "Галактика" західних методик і національної бізнес-практики, швидка адаптація до змін умов бізнесу надають підприємствам-замовникам гнучкий інструмент для вирішення поточних і стратегічних управлінських завдань.

Система "Галактика" дає змогу вести бухгалтерський облік і формувати фінансову звітність, що відповідає вимогам міжнародних стандартів фінансової звітності - International Accounting Standard і загальноприйнятим обліковим принципам -- General Accepted Accounting Principles.

Завдяки модульній структурі системи "Галактика" підприємство-замовник замовляє тільки потрібні йому компоненти системи. При цьому з розвитком бізнесу і появою нових обліково-управлінських завдань, підприємство має можливість послідовно проводити закупівлю необхідних компонент системи.

Апробовані технології розгортання проектів автоматизації сприяють тому, що впровадження системи проходить у стислі терміни, з фіксованим бюджетом і мінімальними для підприємства ризиками. Це дає змогу підприємствам-замовникам швидко окупити витрати на ІТ.

Рішення на базі системи "Галактика", адресовані великим і середнім компаніям, мають різну вартість і відповідають можливостям різних категорій підприємств.

*Business Control* - високотехнологічна ERP-система, що об'єднує найсучасніші вітчизняні і зарубіжні програмні компоненти. Платформа Business Control містить оперативне середовище розробки, завдяки якій кожна компанія може реалізувати власні унікальні вимоги до автоматизації бізнес-процесів. Ядро системи забезпечує швидке формування потрібних

користувачеві баз даних, створення облікових схем, налагоджування бізнес-логіки для кожного конкретного підприємства, складання аналітичних звітів.

Система управління підприємством Business Control дає змогу не тільки акумулювати і зберігати інформацію в єдиній базі даних і оперативно одержувати необхідні відомості за запитом, а й сприяє використанню інформації для аналітики та планування. Система забезпечує повний контроль за діяльністю всіх підрозділів організації і відповідає потребам бізнесу середнього сегмента ринку - від малих до великих компаній.

Готові рішення системи управління підприємством Business Control включають такі модулі: управлінського обліку; управління фінансами; бюджетування; контролю над витратами; взаєморозрахунків; кадрового обліку; автоматизації складського обліку; автоматизації обліку договорів; управління продажем; управління постачанням; управління проектами; автоматизація групової роботи; управління відносинами з клієнтами; інтеграції з іншими програмами (експорт/імпорт відомостей з системи в систему).

Business Control надає широкі можливості для аналітики. У системі реалізовано два підходи до побудови звітів: класичний фінансовий і комерційний. Система Business Control - зручний інструмент для управління компанією, включаючи облік, планування, аналіз і контроль. Business Control - це зручний інтерфейс, висока швидкість роботи, виняткова надійність, простота освоєння й використання, легкість установки та обслуговування.

Система управління підприємством Business Control може поставлятися як повним комплектом, так і конфігураціями, що складаються з певного набору підсистем. Конфігурації призначені для автоматизації окремих ділянок діяльності компанії (управління кадрами, фінансами, торгівлею, клієнтами тощо) та оптимальні для вирішення різних завдань бізнесу.

Система **Віртуоз** - це ERP-система, призначена для середніх і великих промислових підприємств, що забезпечує рішення типових виробничих задач у рамках стандарту MRP II. Система Віртуоз має унікальні за своєю технологічною реалізацією і можливостями рішення для побудови довільних інформаційних моделей корпорацій, які дозволяють оперативно контролювати обстановку у структурних підрозділах.

Система Віртуоз забезпечує рішення таких управлінських задач:

- о організація обліку, аналізу і планування на основі багатовимірної аналітики;

- о створення повного логістичного і фінансового циклів підприємства;

- о ведення складського обліку (управління складом, закупівлями, продажем, облік замовлень/резервувань, автоматичне складання замовлень);

- о розрахунок з дебіторами/кредиторами (управління знижками, ведення прайс-листів, облік бартерних операцій, використання векселів у розрахунках, управління договорами, облік прострочених платежів);

- о ведення обліку основних засобів, малоцінних активів;

- о ведення обліку заробітної плати;

о здійснення касових і банківських операцій, зокрема інтеграції з системами клієнт - банк;

о здійснення податкового обліку;

о планування й управління виробництвом (рішення типових виробничих задач, орієнтація на рішення повного циклу виробничих і логістичних завдань на всіх етапах функціонування виробничого процесу);

о управління документообігом компанії;

о побудова корпоративних моделей компанії. Функціональне призначення: бухгалтерський і податковий

облік, планування, бюджетування, прогнозування, системи управління відносинами з клієнтами, управління документообігом, управління витратами, управління виробництвом, управління постачанням і збутом, управління фінансами" управлінський облік.

**Віртуоз™** - ERP-система з розвиненою базовою функціональністю. Система забезпечує повний логістичний і фінансовий цикли підприємства, допомагаючи в рішенні зв'язаного ланцюжка задач: планування - управління - облік аналіз.

Віртуоз™ - управлінська система, результатом впровадження якої є можливість оперативного отримання необхідної управлінської, інформації, призначеної для внутрішнього використання менеджерами всіх рівнів управління. При цьому автоматично реалізуються всі, необхідні підприємству, облікові функції і прискорюється підготовка зовнішньої звітності для державних органів.

Віртуоз™ - система обліку багатомірних аналітичних об'єктів. Будь-який економічний показник підприємства може бути одержаний у будь-якому аналітичному зрізі. Система спроектована із застосуванням концепцій багатомірного аналітичного обліку інвентарних об'єктів і багатомірного аналітичного фінансового обліку.

Віртуоз™ - інтегрована ERP-система. Будь-яка інформація вводиться в систему тільки один раз, внаслідок чого вона стає доступною для всіх робочих місць, причому в тій формі, яка використовується на кожному з цих робочих місць. Це документарна система. Всі господарські операції проводяться через документи і всі зміни стану об'єктів господарювання відбуваються у результаті виконання господарських операцій.

Віртуоз™ - система моделювання господарських процесів. Гнучка система проведення документів дає змогу одержувати як поточну, так і прогнозну інформацію. Можливе повне і часткове анулювання документів, що дає можливість моделювати господарські процеси і контролювати зміну фінансового стану підприємства.

Це основа для побудови розподіленої інформаційної системи корпорації. Система підтримує єдиний інформаційний простір корпорації на основі власного унікального методу об'єктної реплікації розподілених баз даних. Консолідація даних здійснюється на рівні окремих господарських операцій шляхом передачі первинних документів. При цьому використовується будь-який канал передачі даних. Передача в корпоративну



базу повної аналітичної деталізації операцій кожного підприємства дозволяє одержувати корпоративну звітність у різних аналітичних зрізах, а також надає можливість прослідкувати історію формування того чи іншого показника аж до конкретної господарської операції на конкретному підприємстві.

Це також рішення для роздрібної, дрібнооптової та оптової торгівлі. Робоче місце касира (POS, Point of Sale) забезпечує автоматизацію типових функцій роботи магазину, торговельного залу, супермаркету, роздрібного для опту складу - магазину типу Cash & Carry.

Віртуоз™ - система, що підтримує віддалений доступ. Система дозволяє здійснювати повноцінний доступ до робочого середовища через телекомунікаційні мережі, а не тільки з локальної робочої станції.

Віртуоз™ - це система планування ресурсів підприємств, що реалізовані в архітектурі клієнт - сервер. Система створена з використанням новітніх технологій обробки даних. Архітектура клієнт - сервер забезпечує всю обробку даних на спеціалізованому сервері, а на робочу станцію (клієнт) передається тільки результат запиту. Така архітектура значно підвищує ступінь надійності зберігання й обробки даних, забезпечує надійний захист від несанкціонованого доступу і має потужні засоби з управління розподіленими на кількох серверах даними. Віртуоз™ - гнучка система, що легко адаптується, з кількома рівнями налаштувань.

Це потужний, інструментальний засіб для створення власних економічних моделей. За допомогою інструментальних засобів підприємство може самостійно забезпечити налагодження й модифікацію системи для підтримки власної функціональності. Використовуючи репозиторій - логічний опис

Про - користувач одержує зручний інтерфейс звернень до об'єктів.

**infor:COM** - повномасштабна система класу ERP для підприємств середнього розміру, що забезпечує спільну роботу всіх функціональних підрозділів підприємства, надаючи необхідну налагодженість у роботі підрозділів.

Система об'єднує в єдиний інформаційний простір підприємств модулі планування ресурсів підприємства, планування виробничих ресурсів, управління відносинами з клієнтами, електронної комерції й управління ланцюжками постачань.

Базовий пакет забезпечує функції з ведення продажу, закупівлі, складування, складських операцій виробництва, планування ресурсів тощо. У системі infor.COM можна відстежувати всю історію змін комерційної пропозиції.

Замовлення на продаж можна використовувати для створення виробничих завдань або спеціалізованих замовлень на ремонт чи сервісне обслуговування. Система дає змогу контролювати процес виконання замовлень.

Розрахунок потреб, планування виробництва, закупівлі може здійснюватися по всьому ланцюжку постачань. Модуль "Склад" підтримує

всі складські операції. Система дозволяє здійснювати детальне планування всіх ресурсів. Зворотне, пряме і проміжне планування допомагають контролювати виконання замовлень і виділяти виробничі ресурси з урахуванням точної інформації про завантаження персоналу і матеріалів. За допомогою системи info:COM можна скоротити час виконання замовлень і досягти вищої продуктивності праці.

Модуль "Планування" автоматично формує пропозиції щодо замовлень на закупівлі для модуля "Закупівлі" або робочі наряди для модуля "Виробництво", здобуваючи поточні дані із замовлень, інформації про складські запаси, замовлень на закупівлі і робочих нарядів.

Модуль "Розрахунок собівартості" дозволяє у будь-який час обчислити всі фактичні витрати, що належать до замовлення або проекту, а також відхилення від планових витрат.

У сфері виробництва система дає змогу контролювати випуск продукту, управління виробництвом, облік витрат сировини, процес виготовлення. Передбачені також засоби управління робочими потужностями, термінами і трудовими ресурсами, розширені можливості планування й аналізу, надання управлінської інформації в максимально наочній формі.

Система info:COM спроектована для роботи на базі платформ Microsoft SQLServer, Oracle, або DBII (AS400).

**Smart RetailSuite™** - тиражоване рішення на платформі Microsoft Dynamics NAV, призначене як для окремого магазину, так і для роздрібною мережі, що автоматизує в рамках однієї системи всі бізнес-процеси. Рішення Smart RetailSuite може використовуватися в різних напрямках роздрібного бізнесу - в продовольчих магазинах, універсамах, бутиках, кафе, ресторанах тощо.

Smart RetailSuite підтримує всі стандартні вимоги до систем роздрібною торгівлі, включаючи зміну цін, знижки, зміну кількості, сканування штрих-кодів, пошук необхідної продукції тощо.

Реплікація даних у Smart RetailSuite забезпечує централізоване зберігання інформації в системі для управління пов'язаних мережею магазинів і POS-терміналів. Завдяки рішенню забезпечується управління й обмін всією інформацією між магазинами і центральним офісом.

**Mobile Client** - модуль Smart RetailSuite, який забезпечує роботу терміналів збору даних у режимі реального часу, автоматизує торговельні і складські операції. Mobile Client дає змогу вирішувати такі процедури, як інвентаризація, приймання і відвантаження товару, розміщення і маркування, перевірка і порівняння цін, друк етикеток, управління поверненням товару" віддалений і мобільний продаж.

**SyteLine ERP.** В умовах сучасної економіки підприємства потребують впровадження сучасних інформаційних технологій, що ведуть до підвищення продуктивності і прибутковості бізнесу, забезпечення прибутків і лідерства в конкурентній боротьбі.

Система SyteLine ERP підтримує концепцію обліку потреб клієнтів в основних виробничих і господарських процесах підприємства, обліку переваг клієнта при підборі продукції, і послуг у поєднанні з організованим оперативним управлінням виробництва.

Система SyteLine ERP є гнучкою платформою для розширення за рахунок додаткових комплементарних програмних продуктів, що забезпечують широку функціональність і підтримку новітніх технологій, які дають змогу ефективно управляти бізнесом у різних галузях промисловості.

Система SyteLine ERP має повний набір функцій для забезпечення роботи в таких функціональних сферах як обслуговування клієнтів, управління витратами, обробка замовлень, управління якістю, прогнозування й управління збутом, управління запасами і закупівлями, планування потреб у матеріалах, управління виробничими процесами, планування й управління виробничими потужностями, управління цехами, бухгалтерський і податковий облік, управління персоналом, фінансами, проектами.

У системі SyteLine ERP реалізована підтримка бази постачальників, що допомагає оперативно виявити постачальників за цінами, якістю послуг і термінами постачання.

Варто назвати такі характеристики SyteLine ERP: ергономічний інтерфейс користувача; можливість управління територіально-розподіленими майданчиками, повнофункціональна головна книга, технічна документація он-лайн, підтримка прийняття управлінських рішень, управління документообігом і повідомленнями, високий рівень захисту інформації і розмежування доступу.

SyteLine ERP дає змогу відстежувати виробництво за замовленнями, виробничими графіками або етикетками KANBAN.

SyteLine ERP дозволяє збільшити продаж, автоматизувати бізнес-процеси, підвищити конкурентоспроможність, організувати ефективний логістичний ланцюжок, організувати ефективну взаємодію з клієнтами, оцінювати замовлення точно і швидко, скоротити час виконання замовлення, зменшити операційні витрати, підвищити ефективність праці фахівців, оптимально планувати і використовувати виробничі потужності, зменшити комерційні витрати і собівартість, мінімізувати повернення продукції, оцінювати ступінь досягнення поставленої мети, максимізувати темпи зростання, підвищити рентабельність.

За допомогою системи SyteLine ERP можна швидко обробляти замовлення клієнтів, до того ж, система дозволяє обробляти непрогнозовані, швидкозмінні запити клієнтів.

SyteLine ERP взаємодіє з багатьма системами, зокрема такими, як електронні торговельні майданчики, автоматизовані системи збору даних, система розрахунку заробітної платні, система управління персоналом, генератори звітів, системи планування маршрутів і оптимізації доставки, системи управління документообігом, системи електронного обміну даними, системи управління життєвим циклом продукту (Product Life Cycle), управління відносинами з клієнтами (Customer Relationship Management),

синхронного планування й оптимізації (Advanced Planning & Scheduling - APS), управління ланцюжками постачань (Supply Chain Management - SCM).

Система забезпечує вирішення повного спектра фінансово-економічних завдань, включаючи фінансовий облік, управлінський облік і фінансовий менеджмент.

SyteLine ERP дозволяє приймати від виробничих модулів всю необхідну інформацію про виробничо-господарську діяльність, формувати фінансові показники на основі виробничих даних, консолідувати фінансову інформацію в різних розрізах відповідно до структури бізнесу, забезпечувати усю необхідну звітність.

Забезпечуючи розподілену роботу співробітників різних служб підприємства, SyteLine ERP гарантує безпеку даних. SyteLine ERP відповідає сучасним промисловим стандартам, а також підтримує сучасні комунікаційні стандарти, включаючи XML, архітектуру Microsoft.NET, об'єктно орієнтовані компоненти і 32-бітові клієнтські застосування Windows.

Багаторівнева, клієнт-серверна архітектура SyteLine забезпечує гнучкість конфігурації, сприяючи ефективному зростанню технологічності інформаційного забезпечення підприємства.

**Microsoft Navision** - новітня технологія управління підприємством від провідного світового виробника програмного забезпечення, що охоплює всі аспекти діяльності компанії і дає змогу швидко підвищити ефективність бізнесу, контроль поточних бізнес-процесів, удосконалити бізнес-співпрацю з клієнтами і партнерами.

Microsoft Navision - комплексна інтегрована система призначена для автоматизації всіх видів господарської діяльності підприємства, сферою якої є управління фінансами (FM); управління ланцюжками постачань (SCM); управління відносинами з клієнтами (CRM), управління виробництвом, управління проектами, управління персоналом (HRM), електронна комерція.

#### **6.4.2. Основні модулі ERP-системи компанії BAAN IV**

**Моделювання підприємства** сприяє скороченню термінів впровадження, зниженню рівня витрат і прискореному поверненню вкладених коштів. Процес впровадження починається з опису або розгляду відповідною типу і профілю підприємства референтної моделі. На наступній стадії проводиться коригування параметрів бізнес-моделі з урахуванням вимог замовника. Далі система конфігурується і для кожного конкретного користувача створюється меню, в структуру якого можуть бути включені інструкції і нормативні документи, що визначають виконання окремих завдань. У завершенні проводиться аналіз діяльності підприємства, на основі якого формуються рішення щодо модернізації виробництва, визначаються подальші напрями розвитку.

**BAAN-виробництво** включає планування потреб, конфігу-ратор продукції, управління проектом, управління серійним виробництвом і виробництвом щодо окремих замовлень, управління ланцюжком постачань на рівні корпоративного виробництва. Підсистема "Виробництво" спроектована для роботи з усіма типами стратегій управління виробництвом.

Більш того, система ВААН володіє гнучкістю, що дає змогу змінювати стратегію протягом життєвого циклу проекту.

**Підсистема "Виробництво"** надає також можливість зміни положення точки прив'язки замовлення клієнта, яка визначає ступінь впливу замовлення клієнта на виробничий цикл. Ядром підсистеми "Виробництво" є модуль "Основний виробничий план-графік". Він спроектований для того, щоб допомагати у щоденному управлінні виробництвом водночас із проведенням довгострокового планування й ухваленням рішень. Підсистема дозволяє реалізувати всі типи виробничого середовища й їх поєднання.

**ВААИ-процес** розроблений спеціально для таких галузей промисловості, як хімічна, фармацевтична, харчова і металургійна, і підтримує виробничий процес від досліджень і розробок аж до виробництва, постачання, продажу, збуту і транспортування. Підсистема однаково потужно працює як у рамках окремого підприємства, так і в межах холдингу з територіально розподіленими підприємствами. Ця підсистема повністю інтегрована з усіма іншими підсистемами ВАА>І.

**ВААМ-фінанси** є системою управлінського і фінансового обліку для підприємства будь-якого типу організаційної структури. Система ієрархічних зв'язків робить доступ до інформації й її обробку зручнішою, забезпечує максимально можливу гнучкість при структуризації необхідної інформації. Багато-ланкова структура управління сприяє проведенню аналізу даних головної книги, дебіторської і кредиторської заборгованостей та іншої інформації як на рівні окремого підрозділу, так і на рівні всієї компанії.

Підтримуються три типи календарів: фінансовий, податковий, звітний. У кожному календарі передбачено можливість гнучкого налагодження часових періодів (квартал, місяць, тиждень), що дозволяє фіксувати щоденні операції в межах одного календаря і в той самий час готувати дані для оподаткування в рамках іншого. Підсистема допомагає створювати документацію різними мовами і здійснювати процедури фінансових операцій з необмеженою кількістю валют в умовах різних країн, наприклад, оплату чеками (варіант США і Англії), векселями (Франція), банківськими дорученнями, а також за допомогою електронних засобів.

Модуль ВААМ-збут, постачання, склади забезпечує управління продажем і закупівлею, контрактами, матеріальними запасами і зберіганням, багаторівневе управління партіями і відстеження руху партій товару. Крім цього модуль пропонує управління зовнішньою логістикою і транспортуванням, забезпечує оптимізацію маршрутів, управління замовленнями на транспортування і підтримку транспортних робіт, підтримку загального складування й управління пакувальними роботами. Підсистема "Збут, постачання, склади" відповідає за матеріально-технічне забезпечення виробників і оптовиків. Підсистема повністю інтегрована з усіма продуктами сімейства ВААІЧ, включаючи "Виробництво", "Проект", "Сервіс", "Транспорт" і "Фінанси", що надає доступну й єдину інформаційну систему управління. Ця, повністю інтегрована, система матеріально-

технічного постачання включає електронний обмін даними і зв'язок із плануванням потреб розподілу.

Модуль **ВААИ'Проект** призначений для процедур, пов'язаних із розробкою і виконанням проектів, а також підготовкою комерційних пропозицій для участі в тендерах, і сприяє підвищенню ефективності роботи. Проект забезпечує всі етапи розробки і здійснення проектів, а також підготовки контрактів, включаючи попередню оцінку проектів, укладання контрактів, складання бюджетів, планування, контроль за здійсненням проектів, а також гарантійне і післягарантійне обслуговування. Система автоматично складає замовлення на закупівлю, виробництво необхідних для здійснення проектів виробів, транспортування, має засоби контролю платежів. Це потужний інструмент контролю витрат і доходів, гарантія дотримання термінів постачань. Використання цього модулю дає змогу прогнозувати вплив конкретних проектів на виробничий потенціал і фінансовий стан компанії, що сприяє підвищенню продуктивності праці і використанню наявних ресурсів.

Модуль **ВААЛ-адміністратор** діяльності підприємства є інструментарієм для вдосконалення фінансово-господарської діяльності і розроблений для отримання достовірної інформації за всіма напрямками діяльності компанії. Форма презентації даних дозволяє проводити швидкий аналіз для прийняття безпомилкових рішень.

Модуль **ВААИ-транспорт** створений для компаній, що займаються зовнішнім матеріально-технічним забезпеченням і транспортуванням. Пакет розроблений для всіх видів і модифікацій перевезень і має потужні модулі для управління складами загального користування та упакуванням. Цей блок також може бути конфігуровано відповідно до вимог підприємства. Завдяки своїй гнучкості, підсистема "Транспорт" відповідає різним запитам замовників.

Модуль **BAAN service** призначений для організації управ\* ління всіма видами послуг. Він повністю відповідає вимогам компаній, що виконують післяпродажне і спеціалізоване обслуговування, а також підрозділів, що відповідають за обслуговування усередині підприємства.

Підсистема підтримує всі види обслуговування: періодичне (виконання регламентних робіт і проведення планово-запобіжних заходів), за викликом (ремонт і усунення несправностей при виникненні аварійних ситуацій), наприклад введення в дію об'єктів обслуговування.

Всі дані за місцем розташування устаткування, клієнтів, а також за контрактами на обслуговування і супровід, доступні в оперативному режимі і реєструються для кожного компонента об'єкта обслуговування. Всі види обслуговування можуть виконуватися з урахуванням гарантійних зобов'язань.

**IFS Applications.** Сфера застосування: повнофункціональна ERP система для комплексної автоматизації середніх і великих підприємств. Підтримує управлінські концепції: ERP, EAM, MRP OI, CRM, SCM, PLM, eBusiness.

Виробником є шведська корпорація Industrial&Financial Systems є одним з найбільших європейських розробників програмного забезпечення. Протягом вже 20 років IFS розробляє IFS Applications - комплекс інтегрованих застосувань для управління бізнесом. На розробку IFS Applications витрачено вже близько 600 тис. людино-годин.

У світі IFS Applications використовують більше 3000 компаній. За оцінкою аналітичної компанії Gartner Group, система IFS входить у п'ятірку світових лідерів, у Східній Європі - в трійку лідерів. Gartner Group вважає, що IFS конкурує з SAP за світове лідерство в автоматизації фондомістких галузей.

Функціональні можливості системи такі: обслуговує всі типи виробництва; технічне обслуговування і ремонти обладнання; весь відомий діапазон моделей управління; об'єктно орієнтований підхід, технології XML, PL/SQL, Java, NET, СУБД Oracle, десятки мов програмування.

Користувачі: Volvo, BMW, MG Rover, Land Rover, Lotus, London Taxis, BBC і ВМФ Норвегії, Міноборони Великобританії, ВС Франції і Греції, NEC, Ericsson, Philips Semiconductors, Maxwell.

У 2003 р. IFS Applications стала першою системою, в якій було реалізовано управлінську концепцію 3LM, управління життєвими циклами продукції (PLM), основних фондів (КАМ) і роботи з персоналом і яка включає такі основні модулі, призначені для автоматизації управління підприємством і функціонують в єдиному інформаційному просторі: управління фінансами, документообіг, бюджету на пня, моделі обліку, управління закупівлями, запасами і продажем, основні фонди, облік спеціальних активів, управління персоналом (HRM-система), кадровий облік, розрахунок заробітної плати, управління виробництвом (MRP II), управління витратами, маркетинг і менеджмент (CRM-система).

Система **Millennium ERP** призначена для планування ресурсів підприємства, реєстрації даних про факти фінансово-господарської діяльності й аналізу відхилень. Впровадження Millennium ERP дає змогу значно піл нищити керованість підприємства, і, як наслідок, конкурентоспроможність та прибутковість бізнесу.

Система Millennium ERP має такі основні характеристики; відповідність специфікації MRP II, можливість гнучкого налагоджування бізнес-процесів, генератор звітів, модульність, Па гатоланкова архітектура, повнофункціональна робота мерсі Internet або будь-які модемні з'єднання, підтримку промислових СУБД - Oracle і Interbase, реалізація відкритих стандартів і багатоплатформеність.

Система Millennium ERP забезпечує функціональних завдань відділів:

о відділів збуту - здійснювати планування і контроль відпускання готової продукції і товарів, підвищити якість роботи з клієнтом за рахунок своєчасної обробки замовлень, формувати гнучку цінову політику;

о відділів продажу - одержувати повний обсяг інформації про ринок, потенційних клієнтів і клієнтів, підтримувати історію контактів з кожним клієнтом, автоматизувати рутинні процеси, пов'язані з продажем;

о відділів постачання - здійснювати планування і контроль постачання матеріальних цінностей, стежити за виконанням зобов'язань постачальників і умов;

о складів - контролювати стан складських запасів і рух матеріальних цінностей, резервувати матеріальні цінності для внутрішніх служб і покупців, нормувати запаси матеріальних цінностей на складах, мінімізувати затовареність складів, уникнути фактів розкрадання;

о фінансового відділу - вести оперативний облік і здійснювати контроль фінансових потоків, аналізувати структуру надходження і витрачання засобів, контролювати виконання різних бюджетів;

о фондового відділу - здійснювати планування і контролювати виконання зобов'язань за договорами, вести реєстр договорів з електронними копіями оригіналів;

о бухгалтерії готувати внутрішню і зовнішню звітність відповідно до діючих нормативів з урахуванням особливостей ведення обліку, а також за необхідності ведення бухгалтерського обліку відповідно до міжнародних стандартів фінансової звітності.

Особливості, закладені в документообіг цієї системи, дозволять не тільки налагоджувати порядок обробки документів, але створювати і використовувати власні правила та алгоритми обробки документів і інформації.

У системі реалізована концепція Workflow, що забезпечує повну або часткову координацію виконання операцій (завдань, робіт, функцій), з яких складаються структуровані бізнес-процеси підприємства.

Система аудиту надає можливість протоколювати дії користувачів при роботі з базою даних системи або виконання етапів бізнес-процесів.

Вбудований генератор звітів системи дозволяє створювати або змінювати будь-яку форму звіту чи первинного документа. При цьому формою подання даних може бути не тільки текст чи таблиця, але і діаграма або графік.

Наявність модуля адміністрування дозволяє гнучко здійснювати всі необхідні процедури з обслуговування системи, зокрема розподіляти права доступу користувачів до розділів інформації і порядку виконання етапів бізнес-процесів. Можна використовувати такі модулі.

**Модуль "Планування й управління виробництвом"** є багатофункціональною підсистемою, призначеною для забезпечення планування виробництва і виробничого обліку на підприємстві в рамках специфікації MRP II.

**Модуль "Управління постачанням, збутом, складами"** призначений для вирішення завдань управління матеріальними потоками підприємства, пов'язаних з постачанням, зберіганням, внутрішнім рухом і відвантаженням матеріальних цінностей.

**Модуль "Управління договірними відносинами"** призначений для вирішення завдань планування, обліку і контролю виконання договірних зобов'язань.



**Модуль "Управління розрахунками з партнерами"** призначений для вирішення завдань, пов'язаних з оперативним плануванням і контролем за станом розрахунків із зовнішніми контрагентами підприємства.

**Модуль "Облік зарплати"** є багатфункціональною підсистемою, призначеною для забезпечення своєчасного, повного і достовірного відображення фактів нарахування та утримання заробітної плати з урахуванням специфіки підприємства.

**Модуль "Кадровий облік"** є також багатфункціональною підсистемою, призначеною для ведення кадрового обліку на підприємствах, з централізованим механізмом наказів, що охоплює всю підсистему управління персоналом.

**Модуль "Табельний облік"** є багатфункціональною підсистемою, призначеною для забезпечення своєчасного, повного і достовірного обліку відпрацьованого часу, часу знаходження співробітника у відпустках, на лікарняних, у відрядженнях тощо.

**Модуль "Бухгалтерський облік"** призначений для забезпечення своєчасного, повного і достовірного відображення фактів фінансово-господарської діяльності підприємства за правилами бухгалтерського обліку.

**Модуль "Фінансовий облік"** призначений для забезпечення своєчасного, повного і достовірного відображення фактів фінансової діяльності підприємства за правилами фінансового обліку, визначеними безпосередньо самим підприємством, відповідно до власних потреб і вимог.

**Модуль CRM** призначений для збору, обробки й аналізу даних щодо клієнтів підприємства, зберігання історії контактів з будь-якими зовнішніми контрагентами.

**Модуль "Платіжний календар"** є багатфункціональною підсистемою, призначеною для забезпечення планування і здійснення контролю за грошовими коштами.

**Модуль "Штрихкодування"** призначений для вирішення завдання автоматичного заповнення специфікацій документів на основі зчитування даних з використанням сканерів штрих-кодів. Журнал платежів дає відповідь на питання: які з товарних документів сплачені в певний період часу?

Модуль "Графіки виконання зобов'язань" є багатфункціональною підсистемою, призначеною для забезпечення планування і здійснення контролю виконання зобов'язань за поставаннями, відвантаженнями, оплатою.

**ERP "NOVA"** забезпечує повну автоматизацію управління підприємством, автоматизацію торгівлі. Функціональність забезпечують такі блоки: управління фінансами; управління складом; управління виробництвом; аналітика; адміністрування.

Гнучкий план рахунків, який обирає клієнт (GAAP, IFS, будь-який інший), жодних обмежень в управлінському обліку, забезпечено налагоджування і реалізацію будь-яких кореспонденцій, наприклад, облік при поставаннях і/або транспортних витратах, митних зборах, акцизах, ПДВ тощо. Передбачено додатково до типових проводок опис будь-яких проводок

і виконання будь-яких операцій за рахунками. Відмітні особливості: управління бюджетом - це можливість автоматизованого формування бюджету за минулими періодами, контроль виконання.

Управління складом - це всі операції складського обліку. Відмітні особливості: управління розміщенням товару на складах (на стелажах), сканування документів на товар зі зберіганням у БД (посвідчення якості, сертифікати тощо).

Передбачено оформлення клієнтських замовлень через Web-інтерфейс, наприклад для віддаленої роботи торгових представників. Управління виробництвом - це облік дискретного виробництва. Передбачено складання калькуляцій собівартості з можливістю указання кореспондуючих рахунків.

Реалізований у NOVA модуль "Планування виробництва" дозволяє планувати виробництво у бригадах і контролювати виконання планів. Забезпечується аналітика дійсних витрат сировини і комплектуючих.

Окрім локальних планувань виробництва і закупівель у NOVA передбачено інтегроване планування. Ця функція дає змогу планувати виробництво, закупівлю сировини, матеріалів і готових товарів, контролювати виконання планів.

Підтримується докладна аналітика щодо всіх складських, торгових і фінансових операцій із різними розрізами, контроль відстрочень оплат і прострочених платежів.

Адміністрування дозволяє: описувати користувачів і їх повноваження, контролювати роботу за журналом, налагоджувати складський облік; описувати структуру компанії (холдингу); управляти торгівлею; описувати план рахунків, підстави операцій і статей витрат; розширювати аналітику тощо.

### **6.4.3. Концепція ERP II**

Поєднання традиційної ERP-системи підприємства з Internet-рішеннями для електронного бізнесу привели до створення нового організаційного та управлінського середовища і нової якості системи. У результаті цього з'явилася концепція систем нового покоління - ERP II.

У 2000 р. Gartner Group запропонувала концепцію ERP II (Enterprise Resource and Relationship Processing, управління корпоративними ресурсами і зовнішніми зв'язками), прихід яких на ринок припадає на 2004 р. Основна ідея цієї концепції - вихід за межі задач автоматизації внутрішніх бізнес-процесів підприємства, співпраця покупців і продавців, спільне створення товарів і послуг бізнес-партнерами.

ERP II - бізнес-стратегія підприємства, що включає певний набір застосувань, які допомагають клієнтам і акціонерам компанії збільшувати вартість бізнесу за рахунок новітніх ІТ,

- підтримки й оптимізації процесів як всередині підприємства, так і ззовні. Як приклад можна навести комплексну інтегровану систему "Техноклас". Система спроектована і розроблена відповідно до концепції ERP II за такими принципами:

о інформаційне забезпечення всіх процесів у їх взаємозв'язку;

о одноразове введення даних у місці їх виникнення з прямим застосуванням у всіх функціональних напрямках;

о застосування єдиної бази даних, правил і процедур;

о можливості безперервного розвитку відносно нових корпоративних управлінських методів і технологій, комунікаційних і технічних засобів, інформаційного і функціонального обхвату.

Поява концепції ERP II викликана тим, що для сучасних ERP-систем став характерний розвиток нових функціональних можливостей, пов'язаний з виходом за традиційні рамки оптимізації й автоматизації транзакційних процесів усередині підприємства (технологій SCM і CRM).

**CRM** (Customer Relationship Management) - це орієнтована на побудову конкурентноспроможного бізнесу концепція і бізнес-стратегія, основою якої є клієнто-орієнтований підхід. Ця стратегія, що ґрунтується на використанні нових управлінських та ІКТ, за допомогою яких компанія збирає інформацію про своїх клієнтів на всіх стадіях життєвого циклу спілкування (залучення, утримання, лояльність), здобуває з неї знання і використовує їх на користь свого бізнесу шляхом побудови взаємовигідних відносин з ними.

Головне завдання CRM-систем - підвищення ефективності бізнес-процесів, зосереджених у відділах маркетингу, продажу, обслуговування у фронт-офісі, спрямованих на залучення й утримання клієнтів. На рівні технологій CRM - це набір застосувань, пов'язаних єдиною бізнес-логікою й інтегрованих у корпоративне інформаційне середовище компанії на основі єдиної бази даних. Спеціальне програмне забезпечення дає змогу провести автоматизацію відповідних бізнес-процесів відділів маркетингу, продажу й обслуговування. На практиці інтегрована система CRM забезпечує координацію дій різних відділів, забезпечуючи їх загальною платформою для взаємодії з клієнтами.

**SCM-система** (Supply Chain Management) - система управління ланцюжком постачальників - інтегрована система планування процесів постачання та управління ними, що забезпечує координацію та контроль діяльності всіх учасників ланцюжка постачань. Управління ланцюжком постачань - це інтеграція та управління всіма підприємствами і видами їх діяльності, що входять у ланцюжок постачань, на основі поділеної співпраці, ефективних бізнес-процесів і спільного використання інформаційних ресурсів з метою створення високоефективних систем формування цінності. Ланцюжок постачань включає управління ІС, пошук джерел, закупівлю, складання календарних планів виробництва, обробку замовлень, управління товарно-матеріальними запасами, складування, обслуговування споживачів після продажу.

При цьому традиційний контур управління, властивий ERP-системі, тепер називають застосуваннями back-office, а зовнішні застосування для підприємства - front-office.

Розширення сфери застосування ERP II виявляється у тому, що нова концепція охоплює також невиробничі сфери. Властиві їй функції відображають тепер і специфіку конкретної галузі або напрямку діяльності.

На думку аналітиків Gartner Group, Web-орієнтована архітектура ERP II-систем значно відрізняється від архітектури ERP-систем. Тому для переходу від концепції ERP до ERP II потрібна серйозна перебудова застосувань. Дані, що використовуються в ERP II-системах, відрізняються від внутрішніх даних ERP-систем передусім тим, що вони розраховані на використання в розподіленому торговельному Web-співтоваристві. Крім того, ERP II-системи повністю вмонтовані в Internet і можуть працювати з даними, розміщеними не у власному репозиторії, підтримувати публікацію/підписку, ініційовану клієнтом або подією, взаємодіяти з іншими застосуваннями, що використовують EAI-адаптери (Enterprise Application Integration) і мову XML.

Отже, вирішальними чинниками в конкурентній боротьбі на світовому ринку в майбутньому стануть: швидкість розширення функціональних можливостей ERP-системи й їх адаптації до потреб клієнта; швидкість запровадження систем; якість робіт щодо розширення можливостей, адаптації і запровадження ERP-системи; можливості виробників забезпечити інтеграцію у своїх системах ERP, CRM, SCM і електронної комерції.

Важливу роль у конкурентній боротьбі відіграє здатність компаній до встановлення партнерських відносин для отримання ринкових переваг.

Назвемо основні тенденції розвитку світового ринку ERP-систем: консолідація розробників ERP-систем, прагнення ERP\* виробників підсилювати свої ринкові позиції через придбання компаній, що володіють необхідними технологіями, інтеграція ERP-систем між собою і з іншим корпоративним ПЗ.

Розвиваються стандарти для інтеграції ERP-систем. Зокрема, промислова група RosettaNet розробила нові стандарти для інтеграції ERP-систем на основі стандартів XML.

**RosettaNet** - стандарт, що базується на використанні XML, орієнтований виключно на інтеграцію ланцюжків постачань промислових компаній (B2B) суміжних галузей промисловості (вертикальний стандарт). Стандарт RosettaNet запроваджено в електронній і напівпровідниковій промисловості.

**Модель RosettaNet** (RosettaNet Implementation Framework

- RNIF) використовується для вироблення рекомендацій щодо інтерфейсних партнерських процесів (Partner Interface Process

- PIP). Визначену в RosettaNet сукупність так званих "сигнальних повідомлень моделі" (Framework Signal Messages) використовують у жорстко заданому форматі. Перехід до використання стандарту RosettaNet вимагає від підприємства проведення ретельної підготовчої роботи (визначення бізнес-процесів і їх узгодження з вимогами стандарту RosettaNet, що пов'язано зі значними витратами).

RosettaNet дає змогу створити "Автоматизований ланцюжок попиту" (Automated Demand-Chain), що базується на стандартних описах товарів/послуг. Така система скорочує кількість неправильно оформлених

замовлень та повернень і містить ефективні інструменти для перевірки товарно-матеріальних запасів, конфігурації та сумісності.

Інший стандарт, OAG (Open Applications Group), є горизонтальним стандартом, що базується на XML і застосовується у всіх галузях. Його можна використовувати у застосуваннях, пов'язаних з ERP і управлінням стосунками зі споживачами (CRM). Результат розробки OAG - уніфікація бізнес-повідомлень і сценаріїв для інтеграції застосувань підприємств. Важливою функцією OAG є можливість додавання окремих бізнес-застосувань без заміни всього комплексу після впровадження OAG.

**Стандарт BizTalk Framework** компанії Microsoft розроблено для підтримки при створенні й обслуговуванні схем даних XML, що забезпечують інтеграцію застосувань електронного бізнесу. Ці схеми доступні усім, хто бажає за їх допомогою забезпечити інтеграцію бізнес-систем з іншими системами, що використовують ті самі схеми. Схеми призначені лише для читання, тому користувач, обравши конкретну схему для інтеграції власних систем, вже не може її змінювати.

У BizTalk організовано спільний репозиторій, де лінійки сумісних з BizTalk форматів можна перевіряти на правильність, зберігати, одержувати і вільно використовувати. Крім того, розробників сумісних з BizTalk стандартами, заохочують надавати конвертори повідомлень у відомих форматах даних. Торгові партнери можуть обмінюватися повідомленнями, які відповідають схемам, що опубліковані в бібліотеці схем BizTalk.

Open-EDI Reference Model (ISO 14662). Завданням специфікації еталонної моделі відкритого електронного обміну даними - національний стандарт ISO 14662 - є подоланням проблем, що виникли при застосуванні технології EDI. Насамперед це вирішення питання узгодження економіко-технічних проблем детальних двосторонніх угод шляхом розробки стандартних бізнес-сценаріїв і допоміжних послуг.

Ця модель вважається базовою для виконання бізнес-транзакцій у межах окремих галузей економіки або між ними.

Специфікації Open-EDI не залежать від конкретних реалізацій, конкретного бізнесу, угод, видів економічної діяльності і підприємств. На перший план у специфікаціях Open-EDI виступають бізнес-аспекти, а не системні характеристики транзакцій.

Для опису відповідних аспектів бізнес-транзакцій Open-EDI Reference Model використовує операційний рівень подання бізнесу, що враховує семантику даних у транзакціях, і пов'язаний з ними обмін даними, а також правила проведення транзакцій. Функціональні служби враховують наявність допоміжних служб, які задовольняють технічні потреби: можливості доставки послуг, інтерфейси і протоколи.

Більшість запроваджень ЕОД (електронного обліку даними) успішні лише у разі довгострокових партнерських взаємовідносин, обмеженої кількості партнерів.

Відкритий ЕОД зменшує дію цих бар'єрів за допомогою впровадження стандартних бізнес-сценаріїв та необхідного сервісу для їхнього

обслуговування. Якщо бізнес-сценарій погоджено, і його реалізація узгоджується зі стандартами відкритого ЕОД, то у попередній угоді між діловими партнерами, окрім рішень щодо транзакцій (операцій) відкритого ЕОД відповідно до бізнес-сценарію, немає необхідності. Оскільки відкритий ЕОД застосовує універсальний підхід, то це дає змогу підприємствам встановлювати короткострокові взаємовідносини швидко й ефективно з *боку* витрат. Бізнес-сценарії та необхідний обслуговуючий (супутній) сервіс буде доступний для всіх, хто бажає їх застосовувати, забезпечуючи таким чином необхідні засоби для запровадження відкритого ЕОД.

Сферою застосування відкритого ЕОД є електронне оброблення бізнес-транзакцій між різноманітними автономними організаціями, органами влади або фізичними особами всередині та між секторами економіки (наприклад, державний (приватний), промисловий або географічний). Застосування охоплює бізнес-транзакції, які містять різні типи даних, а саме: числа, символи, образи та *звук*.

Спочатку еталонну модель відкритого ЕОД було розроблено для впровадження стандартів, необхідних для колективної (сумісної) роботи організацій через ІТ-системи спільного доступу. Ця модель не залежить від:

- о застосування інформаційних технологій;
- о сутності або правил ведення бізнесу;
- о бізнес-діяльності;
- о сторін, що беруть участь у бізнесі.

Еталонна модель відкритого ЕОД визначає необхідні стандарти для відкритого обміну даними й є еталоном цих стандартів внаслідок визначення базових концепцій, застосованих для їх розроблення. Вона є основою для координації роботи між різними агенціями, залученими до стандартизації ЕОД, забезпечує рамки для координації та інтеграції чинних, нових та розроблення майбутніх стандартів. Еталонна модель відкритого ЕОД окреслює перспективу для чинних стандартів з електронного бізнесу.

Для опису відповідних аспектів бізнес-транзакцій еталонна модель відкритого ЕОД застосовує два види подання: бізнес-операційне подання (БОП); функціонально-сервісне подання (ФСП).

БОП призначене для тих аспектів, які застосовують для бізнесових потреб відкритого ЕОД, зокрема це: семантика бізнес-даних у бізнес-транзакціях і пов'язаний з ними обмін даними, правила для бізнес-транзакцій, що містять операційні правила (процедурні домовленості ведення бізнесу), угоди, спільні зобов'язання.

ФСП призначене для супутнього обслуговування, що задовольняє механістичним вимогам відкритого ЕОД. Воно зосереджує увагу на аспектах інформаційних технологій, а саме функціональних можливостях; сервісних інтерфейсах, протоколах.

Такі функціональні можливості, сервісні інтерфейси і протоколи включають можливості ініціації, оброблення і супроводу процесу відкритого ЕОД, інтерфейс застосунку користувача; інтерфейс інфраструктури

передавання; оброблення механізму захисту; протоколи для спільної (колективної) роботи ІТ-систем різних організацій; механізми перекладу.

Провідні постачальники ERP-систем (SAP AG, PeopleSoft, J.D. Edwards) постійно поглиблюють роботу зі створення стандартів і протоколів обміну інформацією і взаємодії ERP-систем та зовнішніх аналітичних застосувань.

Нині швидко розвивається ринок оренди через Internet різних застосувань *ASP* (Application Service Providing/Provision). *ASP* - це технологія використання елементів інформаційних технологій на умовах орендної плати (не плутати з ActiveServer Pages(*ASP*) - технологією Microsoft для Windows систем, що дозволяє створювати динамічні Web-сторінки).

*ASP* підтримує кілька мов програмування, найбільш поширеною серед яких є VBScript. Сервер застосувань і сервер бази даних можуть бути встановлені на устаткуванні *ASP*-провай-дера, тоді користувачі ERP-системи можуть працювати з віддаленим сервером застосувань через Internet (рис. 6.6). При цьому *ASP*-провайдер ERP-системи може спеціалізуватися на вертикальних ринках. У такому разі підприємство дістає доступ до ERP-системи, що орієнтована на певну галузь.

Переваги такого підходу очевидні - можливість доступу до багаторівневих застосувань, надійність їх функціонування. Крім того, виключається необхідність витрат на придбання програмно-апаратного забезпечення, оплату послуг впровадження і заміну ІТ-підрозділу для підтримки ERP-системи. Є різні базові способи доступу до застосувань: термінальний доступ через Microsoft Windows NT Terminal Edition, через стандартний Web-браузер через Internet, тонкий Java-клієнт.

В інформаційних технологіях "тонкий" клієнт - це ком-п'ютер-клієнт мережі з клієнт-серверною архітектурою, який переносить всі задачі щодо обробки інформації на сервер.

Серед *ASP*-застосувань поширені офісні пакети, застосування для електронного бізнесу, корпоративні інформаційні системи, аналітичні програми, системи управління персоналом, застосування управління активами підприємства, життєвим циклом виробу.

Причини застосування *ASP*-моделей такі: дозволяють компаніям знизити накладні витрати, зростає швидкість впровадження застосувань, вирішення проблем відсутності кваліфікованих ІТ-фахівців, з'являється можливість швидкого доступу до нових технологій в Про. Проте є ряд недоліків *ASP*-моделей, а саме: інформаційна безпека, недостатній рівень якості

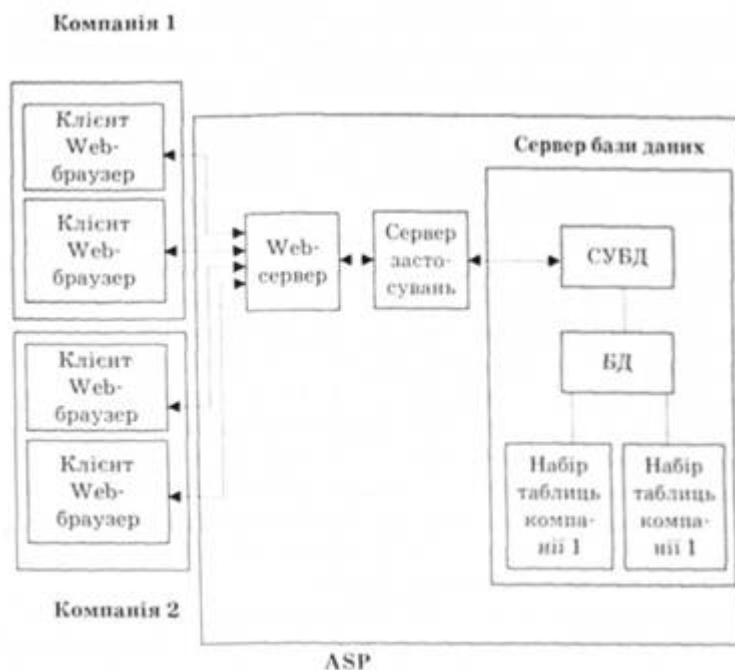


Рис. 6.6. Структура ASP

ASP-сервісу, труднощі забезпечення гнучкості налагоджування ASP-продуктів для потреб підприємства, ризик низької продуктивності системи, пов'язаний з обмеженими можливостями доступу в Internet, вимога до жорсткої регламентації бізнес-процесів підприємства у межах встановлених ASP-застосувань.

ASP-технологія може бути корисною малим і середнім підприємствам, які не можуть надійно прогнозувати оборотні кошти і дозволити собі придбати ERP-систему.

Саме в управлінні постачаннями при використанні Internet, можуть проявлятися всі переваги ASP-технології, оренда застосувань. Це показує досвід США, Великобританії, Німеччини та інших розвинених країн. Перехід на ASP припускає інтеграцію інформаційних ресурсів всіх учасників ланцюжка постачань на єдиній технологічній платформі провайдера ASP-послуг, що обіцяє серйозні конкурентні переваги і забезпечує можливість виходу на нові, передові, форми організації бізнес-процесів на основі партнерських відносин між усіма учасниками мережі постачань, включаючи ASP-провайдерів і кінцевих споживачів. Ця технологія, що реалізовується на основі аут-сорсингу, якнайкраще відповідає сучасним ринковим концепціям SCM (i2, LogiPlan, Manugistics, InterLogistics) і CRM.

Доступ до ІС через Web забезпечує роботу ланцюжка постачань у режимі реального часу, що суттєво підвищує точність і надійність виконання бізнес-процесів, якість проектування логістичних послуг, а також скорочує логістичний цикл. Можливість безперервного контролю бізнес-процесів у ланцюжку і оперативного прийняття управлінських рішень на його основі сприяє зменшенню кількості і впливу збоїв та помилок у виконанні замовлень. При цьому витрати на ASP можуть бути розділені між компаніями, що забезпечують функціонування ланцюжка на принципах аутсорсингу.



Історія розвитку технології ASP як різновиду IT-аутсорсин-гу у сфері систем автоматизації бізнес-процесів підприємства розпочинається з 90-х років XX ст. у СУТ А. Ідея використовувати клієнт-серверні рішення в роботі зі складними інформаційними ERP-системами віддалено через Web-сервер є перспективною. Не дуже високий попит на ASP-рішення, зумовлений складністю, ризикованою і високою вартістю проектів, перешкоджав розвитку ринку і не дозволяв вийти на конкурентний з локальними рішеннями рівень цін. Ця обставина стала однією з головних перешкод, оскільки відплив конфіденційної інформації, знань, оригінальних технологій різко звужували коло бажаючих скористатися ASP.

З остаточним формуванням ринку споживача і затвердженням парадигми загальної співпраці в частині створення ланцюжків вартості, розвитком аутсорсингу і систем управління стосунками з клієнтами (споживачами) інтерес до ASP знову зростає. ASP нині становить частину систем управління бізнесом у Європейському Союзі (табл. 6.2).

**Таблиця 6.2.** Найбільш поширені АЗР-рішення для управління бізнесом компанії в США та Великобританії

Рішення	Функціональність
SAP (mySAP), Oracle Business Suite, JD-Edwards, People Soft, Lawson, Great Plains	Корпоративні інформаційні системи (ERP) для комплексної автоматизації управління бізнес-процесами компаній, які мають у складі пакета контур "Логістика". Системи для великого бізнесу у Web-використанні
Siebel, Sales Logix	Системи CRM-класу для великого та середнього бізнесу відповідно. Автоматизація управління взаємовідносинами з клієнтами (покупцями)
MS CRM, MS Exchange, MS SQL Server, MBS (Axapta, Navision), Great Plains	Лінійка рішень Microsoft. Комплексне (інтегроване) ASP-рішення на основі продуктів одного виробника
Data Center, Data Warehouse, Call Center	Центри і сховища даних для спільного використання постачальниками й споживачами товарів та послуг, центри виклику для роботи із замовленнями і заявками клієнтів

Ризик втрати керованості внаслідок проблем в інформаційному забезпеченні стримує поширення ASP. Більш того, до 35 % угод про передачу IT-функцій компаній провайдерам не поновлюються або припиняються достроково. ASP може бути реалізоване у формі аутсорсингу або звичайної оренди, при якій цінність (value) у покупця цих послуг може і не збільшуватися.

Аутсорсинг починався з передачі функцій створення, впровадження і супроводу інформаційних систем розробнику. Потім він трансформувався в

нову форму, що одержала назву Task Sourcing (TS). Прикладами реалізації TS, зокрема, можна вважати послуги щодо планування оптимальних маршрутів доставки товарів; рішення завдань оптимізації завантаження транспортних одиниць, що надаються інтерактивно через мережу Internet.

Серед ASP-рішень можна виділити програмне забезпечення, яке реалізує функції електронних платежів у системах електронної комерції, електронного магазину, систем дистанційного навчання, поштові й офісні системи, системи захисту даних, інформаційно-правові системи, інформаційно-пошукові системи, ПЗ, що забезпечують функціонування Call Centers, системи автоматизації бізнес-процесів компанії (системи ERP, CRM і SCM - класів).

Є два варіанти реалізації ASP, кожен з яких характеризується особливою організацією використання застосувань; відношення "один до багатьох" - провайдер надає одну програму в розпорядження багатьом клієнтам; персоналізований доступ (відношення "один до одного") - застосування повністю надається одному клієнтові, чим забезпечується можливість гнучкого налагодження під наявні бізнес-процеси.

Перспективи застосування ASP в управлінні ланцюжками постачань очевидні. Про це свідчать відомості із західного ринку ASP-послуг. Зокрема, поява програмних продуктів для управління, орієнтованих на малий і середній бізнес, сприяє активному формуванню ASP-перспективної цільової групи. У вдалих ASP-проектах забезпечується істотна економія ІТ-ресурсів (більш ніж на 20 %) за рахунок зниження сукупної вартості володіння цими ресурсами (TCO - Total Cost of Ownership). Основні переваги ASP-рішень полягають у таких принципах:

- о оперативний доступ до новітніх ІТ-рішень і кращих практик управління бізнес-процесами, реалізованим в інформаційних системах провідних виробників;

- о витрати на управлінське ПЗ, фіксовані на рівні орендної плати. Відсутні непередбачені витрати на супровід та експлуатацію ІС, що підвищує точність прогнозування витрат і сприяє фінансовій стійкості компанії;

- о вивільнення внутрішніх ІТ-ресурсів компанії за рахунок передачі функцій автоматизації управління ASP-провайдерові;

- о низькі стартові витрати на автоматизацію управління;

- о істотне зниження ризику, пов'язаного з вибором і впровадженням ІС;

- о незалежний доступ до ІС за допомогою мережі Internet віддаленого клієнта з будь-якого ПК;

- о можливість зниження бази оподаткування за рахунок віднесення витрат, пов'язаних з ASP, на собівартість товарів чи послуг;

- о забезпечення надійного функціонування ІС. Провайдер має у своєму розпорядженні потужніше устаткування, а також кваліфікованіший персонал, ніж клієнти;

- о можливість реалізації розподіленого офісу для координації роботи всіх учасників ланцюжка постачань через Call Center або Enterprise Data Management (EDM, Центр управління даними підприємства).

Особливо доцільне використання технології ASP на стадіях активного розвитку бізнесу, при реалізації пілотних бізнес-проектів і проектів з коротким життєвим циклом, а також за необхідності впровадження єдиної моделі управління в групі компаній з філіалами і відділеннями.

Найчастіше IT-аутсорсинг використовують великі компанії, що працюють на інноваційних і висококонкурентних ринках. Основним мотивом при цьому зазвичай є бажання різко скоротити кількість забезпечувальних відділів, служб і департаментів, щоб зосередитися на ключових бізнес-процесах компанії і проблемах стратегії. Крім того, витрати на ASP контролювати набагато легше, ніж засоби, що виділяються на підтримку власних IT-структур. Додатковими мотиваційними чинниками можуть бути обмежений IT-бюджет, проблеми із залученням висококваліфікованого IT-персоналу для повно-масштабної автоматизації управління, відсутність часу на створення власних IT-ресурсів, швидке зростання номенклатури товарів і послуг.

Поза сумнівом, вибору ASP сприяє наявність якісного Internet-каналу, безпечні і надійні комунікаційні мережі, а також стандартизація документообігу в компанії й орієнтація на технології електронної комерції. Основною причиною повільного впровадження ASP у практику є:

- о обмеженість вибору постачальників ASP-послуг і як наслідок відсутність достатньо привабливих пропозицій щодо співвідношення ціна/якість;

- о конфлікт інтересів: провайдерів вигідний довгостроковий контракт, а клієнтові навпаки, обумовлений життєвим циклом товару або бізнес-проекту;

- о недостатня зацікавленість аутсорсерів (і провайдерів ASP-рішень) у розвитку відносин довіри з клієнтами, що характерно для вітчизняних бізнес-структур;

- о високі транзакційні витрати й організаційні проблеми в забезпеченні взаємодії з клієнтами, посередниками і постачальниками через IT-провайдера;

- о неадекватна оцінка ризиків при аналізі витрат на IT.

Проблемою на шляху поширення ASP в Україні залишається нерозвинута нормативно-законодавча база, зокрема відсутність єдиних норм регулювання договірних відносин, надання звітності і формалізації контролю за бізнес-процесами на основі SLA і розподілу ризиків. Досить високими залишаються і витрати на високошвидкісні комунікаційні канали та засоби захисту інформації.

До основних статей витрат, пов'язаних з реалізацією IT-проекту на власному майданчику належать: вартість ліцензій на прикладне ПЗ; обслуговування ПЗ (підтримка продавця); навчання персоналу; витрати на придбання нового і модернізацію старого устаткування; вартість системного ПЗ; технічна підтримка, експлуатація і модернізація інформаційної системи; адміністрування БД; мережеве адміністрування й управління комунікаційними каналами; забезпечення інформаційної безпеки (системи

захисту інформації, резервне копіювання, відновлення ІС після збоїв); витрати на розвиток ІС.

Витрати на ASP складаються із щомісячної орендної плати за кожне робоче місце й одноразових витрат на навчання. Причому практикуються знижки на орендну плату залежно від кількості робочих місць і тривалості ASP-контракту. Успішність реалізації ASP-проекту значною мірою залежить від вибору провайдера. При цьому слід звертати увагу на: наявність позитивного досвіду реалізації ASP-проектів; наявність і характер відгуків клієнтів провайдера, організацію референц-візитів; техніко-технологічний рівень підтримки ASP-технологій і адекватність ІТ-інфраструктури провайдера вибраному клієнтом ASP-рішенню; фінансову стабільність провайдера і пов'язаних з ним аутсорсерів ПЗ; достатній рівень компетентності у сфері вертикальних інтегрованих ринків; якісний рівень відносин з постачальниками ПЗ, клієнтами і готовність розвивати клієнтське ПЗ; надання можливості тестування ASP-рішення.

Контракт має містити угоду про рівень сервісу (SLA) і компенсаційну схему на випадок його порушення. В угоді про рівень сервісу зазвичай обумовлюється: термін дії угоди; територія й об'єкти обслуговування, гарантований час відгуку на клієнтські запити; форми звітів про роботу системи; система критеріїв якості послуг; вимога нерозповсюдження конфіденційної інформації.

Прогнозується, що компанії о-розробники ERP-систем поступово перетворяться на фірми з обслуговування застосувань. Відомі ERP-розробники пропонують ASP-варіанти своїх програмних продуктів.

Відбувається перехід від клієнт-серверних ERP-систем до "тонкого" Web-клієнта і підтримка розподілених компонентних технологій проміжного ПЗ типу CORBA.

До недавнього часу в основі ERP-систем лежала клієнт-серверна модель розподілених обчислень з обмеженою кількістю користувачів і встановленим розміром БД. Майбутнє ERP-систем ґрунтується на тісній інтеграції наявних застосувань і бізнес-процесів компаній за допомогою проміжного ПЗ. Однією з ключових технологій у цьому напрямі є мова XML, технології EJB і CORBA, а базовою платформою буде Java, розвиток розробниками ERP-систем CRM і SCM-застосувань, засобів бізнес-аналізу та обміну даними між бізнес-застосуваннями.

Крім того, влітку 2001 р. компанія J.D. Edwards інтегрувала у реальному часі технологію Advanced Planning Solutions (APS) з ERP-системою One World (раніше була реалізована тільки пакетна інтеграція).

Сучасні ІС мають бути Web-орієнтованими, що означає Internet-орієнтованість усіх модулів ERP-системи, підтримувати інтеграцію із застосуваннями електронного бізнесу, подальшу диверсифікацію розробки ERP-систем і поділ праці, глобалізацію бізнесу щодо розробки.

Інтеграція систем ERP із системами електронної комерції B2B і B2C - закономірний етап у розвитку технології управління ресурсами підприємства. Створення й експлуатація систем e-комерції, передусім систем B2B, стає

найефективнішою, якщо ці системи інтегровані в загальнокорпоративні бізнес-процеси і відповідно вбудовані в ERP-систему.

Міжкорпоративна інтеграція на рівні взаємозв'язків між ERP-системами постачальників і споживачів забезпечується через B2B-системи електронної комерції - електронні системи збуту, постачання й електронні торговельні майданчики. Оскільки електронний торговий майданчик як система електронної комерції B2B дозволяє здійснити пряму взаємодію між суб'єктами ринку - постачальниками і споживачами, то вона може стати елементом інтеграції між ERP-системами суб'єктів ринку. У цьому випадку окремі корпоративні системи управління ресурсами стають частиною глобального електронного ринку.

Щоб розвиватися, виробники повинні розробляти нові технології і бізнес-процеси з метою задоволення індивідуальних купівельних потреб споживачів, відповідати на ці потреби товарами і послугами, які становлять унікальну цінність для кожного покупця. Виробники повинні зробити часткову зміну у стратегії й інтегрувати покупця в центр процесу планування діяльності підприємства. Інтеграція покупця з ключовими бізнес-процесами підприємства змінює стратегію розвитку підприємства, тому з'явилася нова модель управління діяльністю підприємства - планування ресурсів, синхронізоване з покупцем (CSRP).

#### **6.5. CSRP-система (планування ресурсів, синхронізоване зі споживачем)**

Система планування виробництва нового тисячоліття буде фокусуватися на виробничій ефективності і на створенні купівельної цінності. СБКР - це перша бізнес-методологія, що інтегрує діяльність підприємства, орієнтовану на покупця, в центр системи управління бізнесом. CSRP нагтпновнолюг методологію ведення бізнесу, що базується на поточній інформації про покупця, й акценти підприємства зміщуються з планування від потреб виробництва до планування замовлень покупців. Інформація про покупців і послуги стає базисом системи підприємства. Діяльність з виробничого планування розтирки ть-ся і замінюється запитами покупців, переданими з підрозділів організації, орієнтованих на роботу з покупцями.

CSRP перевизначає практику бізнесу, фокусуючи її на рим\* копій активності, а не на виробничій діяльності. Віл мес процеси підприємства синхронізуються з поведінкою покупців.

CSRP забезпечує обслуговування покупців і розширим його, виходячи за межі звичайної телефонної підтримки і видачі довідки про рахунки. При використанні моделі CSRP купівельні послуги стають "нервовою системою" підприємства, командним пунктом для організації. Центр технічної підтримки покупців відповідає за доведення інформації про покупців до виконавчих відділів підприємства.

**Концепція CSRP** - концепція управління ресурсами під\* приємства, орієнтована на потреби пілпригмств-спожинячіп. Вона враховує не тільки виробничі і матеріальні ресурси, але ресурси всього життєвого ииклу товару, що мають місце під час маркетингової роботи з клієнтом, післяпродажного

обслуговування. Ця особливість CSRP набуває вирішального значення для підвищення конкурентоспроможності підприємства у галузях, де життєвий цикл товару невеликий і необхідна оперативна реакція на зміну потреб споживача. Щоб управляти вартістю товару, оцінювати його вартість протупаним и обслу-гону впини, необхідно причому мати всі елементи жигтеиого циклу цього товару.

CSRP-система інтегрована електронна інформаційна система управління, що реалізовує концепцію CSRP. Призначення CSRP - створення товарів з іідиіиценою цінністю для покупця, тобто продуктів, які повністю задовольняють специфічні вимоги конкретного покупця.

Реалізація концепції CSRP на конкретному підприємстві дає змогу підприємству ефектинніше працювати, що дозволяє управляти замовленнями клієнтів і всією роботою з ними. Стає можливою погодинна зміна графіка поставок, що в умовах ERP не завжди можливо. Детальний аналіз собівартості замовлення і конкретних елементів у його складі стає можливим вже на етапі його оформлення. При розрахунку собівартості можна врахувати всі додаткові операції з адміністративного обслуговування замовлення.

Основний принцип методології CSRP - в інтеграції системи обробки інформації про покупця в систему процесу планування й управління діяльністю підприємства (рис. 6.7). У результаті цього покупці можуть здійснювати вплив на ключові бізнес-процеси організації, змінювати її стратегію і послідовність дій щодо реалізації цієї стратегії, одже інформація про покупця - ядро в CSRP -системі планування й управління діяльністю підприємства.



**Рис, 6.7.** Інформація про покупця

У рамках CSRP важливого значення набуває інтеграція системи управління ресурсами підприємства з програмними продуктами її

контрагентів, що реалізують специфічні завдання управління (наприклад, управління технологічним обладнанням, проектуванням виробів). Такими продуктами можуть бути, наприклад, системи оптимального розміщення замовлень, управління технологією виготовлення товарів тощо.

Використовуючи застосунки підходу CSRP, продавець має змогу зафіксувати специфічні вимоги до продукту, його ціну та автоматично надіслати цю інформацію в офіс підприємства, де інформація про вимоги до продукту динамічно перетворюється на детальні інструкції з виробництва і планування.

Складається список матеріалів і комплектуючих для виробництва, автоматично визначаються виробничі маршрути, матеріали плануються й замовляються і, нарешті, створюється замовлення. Критична для покупця інформація динамічно інтегрується в основну діяльність підприємства.

Після цього інформація про критичні переваги покупця зберігається в центральній БД про покупців, яку можуть використовувати підрозділи обслуговування покупців, технічного обслуговування, досліджень, планування виробництва тощо. Після цього інформація про критичні переваги покупця зберігається в центральній БД про покупців, яку можуть використовувати підрозділи обслуговування покупців, технічного обслуговування, досліджень, планування виробництва тощо. Діяльність підприємства підтримується через динамічну біз-нес-систему на базі відкритих технологій (Java, XML, Web-сервіси тощо) і синхронізується з потребами покупців.

Покупець використовує браузер для доступу до Web-сервера виробника, щоб ввести замовлення - стандартне чи модифіковане - у будь-який час дня або ночі. Покупець може замінити попередні замовлення, перевірити стан ще не виконаних замовлень або подати новий запит.

Переваги використання систем класу ERP II, CSRP: підвищення цінності продукції за рахунок її персоніфікації у масових масштабах; висока адаптованість до кон'юнктури ринку, яка забезпечується удосконаленням динамічного планування виробництва щодо замовлень у режимі реального часу, підвищення конкурентних позицій, що здійснюється шляхом коор-

динації всіх бізнес-процесів підприємства. Підрозділи планування можуть у високій динаміці змінювати роботи, послідовність виконання замовлень, виконувати вчасно виготовлення товарів, оскільки засоби підтримки покупців об'єднуються з важливими процедурами планування, виробництва й управління (рис. 6.8), а методи, що базуються на новітніх ІКТ, розширюють можливості підтримки покупців, включаючи віддалений і цілодобовий сервіс за принципом самообслуговування тощо.

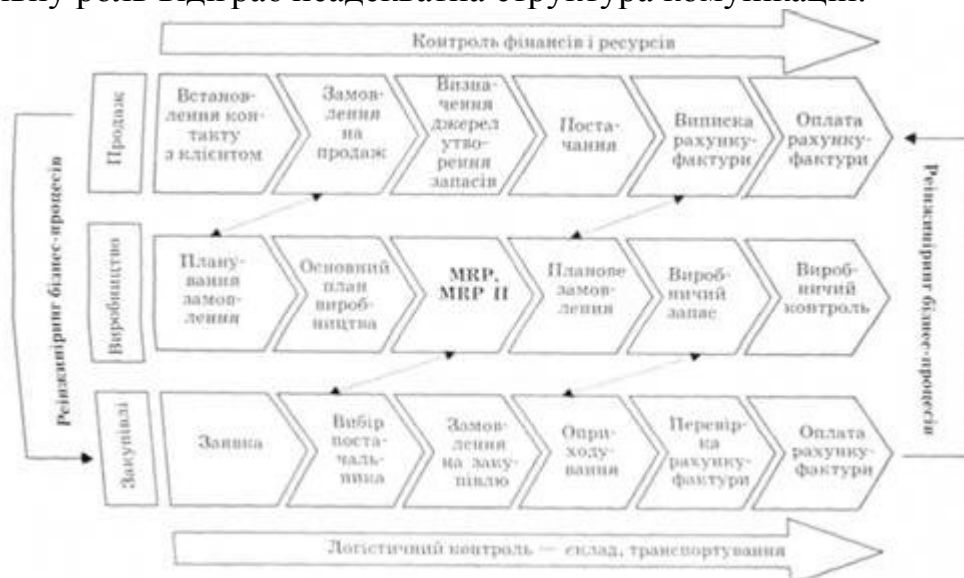
В епоху інформаційної економіки можливість своєчасного отримання точної інформації для бізнесу найважливіша. Персоніфіковане задоволення споживачів головна тенденція роботи компаній.

На зростання важливості інформації в бізнесі впливає три чинники: інформація про стан виконання замовлення, наявність товарів/послуг, календарні графіки поставок і рахунки-фактури. Наявність необхідної

інформації дає змогу керівникам вживати заходи щодо скорочення товарно-матеріальних запасів і співробітників до рівня, який забезпечує конкурентоспроможність фірми; знання про інформаційні потоки важливі для стратегічного, тактичного та оперативного управління, стратегічного планування та ефективного використання ресурсів.

На кожному етапі розвитку організаційної структури підприємства відбувається постійний реінжиніринг: проводиться облік результатів діяльності; проводиться аудит і скорочення витрат, створюється новий продукт, під нього динамічно перебудовується виробництво, з'являється необхідність в інтегрованій функціональній системі керування.

У сучасних умовах методи ведення бізнесу й управління підприємством, побудовані за функціональним принципом, не є ефективними, тому що окремі функціональні підрозділи компанії мають локальне, вузьке бачення проблем, які постають перед підприємством. Ці підрозділи зазвичай не зацікавлені у виконанні робіт, що безпосередньо не входять у їхні функції та обов'язки. Інколи поширюється внутрішня конкуренція, трапляються конфлікти між окремими підрозділами. Крім того, негативну роль відіграє неадекватна структура комунікацій.



Збільшення кількості бізнес-процесів підприємства під час виробництва продукції призводить до ускладнень. Керувати такими процесами стає значно важче. Тому виникає потреба в інтелектуальних автоматизованих інформаційних системах (ІАІС) для підтримки цих процесів.

Завдання, які має вирішувати ІАІС підприємства (рис. 6.9): збір, підготовка, подання, обробка даних; аналітична обробка даних; інтелектуальний аналіз даних; прогнозування, імітаційне моделювання, оперативне та стратегічне планування; синтезоване управління підприємством тощо.





**Рис. 6.9.** Інжиніринг інформаційних потоків

Розвиток підприємства залежить від здатності знаходити споживача товарі в/послуг його діяльності й якісно задовольняти ці запити. В умовах становлення інформаційної економіки клієнти вимагають особливого підходу, при цьому кардинально змінюється сам статус клієнта:

- о працівники підприємства високоосвічені, їхні прагнення і мотивації спрямовані на виконання складніших, інтелектуальних і відповідальних завдань, відбувається безперервне підвищення кваліфікації, професіоналізму працівників;

- о ринок послуг набуває глобального характеру, конкуренція між підприємствами стає більш різноманітною завдяки застосуванню сучасних ІКТ, які є базовими для розвитку, удосконалення і впровадження найважливіших інших технологій.

Переваги використання систем класу CSRP:

- о підвищення споживацької цінності продукції, що досягається шляхом вивчення кон'юнктури ринку;

- о швидка адаптація до кон'юнктури ринку, що забезпечується удосконаленням виробничого планування;

- о зміцнення конкурентних позицій підприємства, що забезпечується координацією бізнес-процесів з покупцями завдяки доступу інформації про замовлення покупців у режимі реального часу;

- о підвищення якості товарів, оскільки засоби підтримки покупців поєднуються з ключовими застосуваннями планування, виробництва та управління. Необхідна інформація про покупців і товари заздалегідь поставляється підрозділам, що відповідають за виробництво, продаж, дослідження і розвиток, а також іншим підрозділам;

о Web-технології розширюють можливості підтримки покупців, включаючи віддалений, цілодобовий сервіс.

Ключові виконавські системи автоматично змінюються, надаючи покупцям відповіді і послуги швидше.

Центри підтримки покупців стають центрами продажу і підтримки користувачів. Інтеграція з продажем, обробкою замовлень та управлінням забезпечує знання й інфраструктуру для перетворення підтримки покупців на діяльність із продажу, забезпечуючи канал для просування нових і супутніх продуктів та послуг.

Планування виробництва і всієї діяльності перевизначається і стає плануванням замовлень покупців і динамічним виробництвом.

Безпосередня інтеграція з Інформацією про конфігурацію замовлень дає змогу виробничим підрозділам забезпечити цілісність процесу планування шляхом зниження кількості повторної роботи і перерв внаслідок напливу замовлень. Удосконалення виробничого планування надає можливість виробникам забезпечити кращу оцінку термінів постачань і поліпшити вчасне постачання.

Виробниче планування дозволяє оптимізувати операції, спираючись на дійсні купівельні замовлення, а не на прогнози чи оцінки. З доступом до точної інформації про замовлення покупців у режимі реального часу підрозділи планування можуть динамічно змінювати групування робіт, послідовність виконання замовлень покупців, придбання, а також висновки суб-контрактів з метою поліпшення обслуговування покупців і зниження вартості.

Вимоги покупців до продукту можуть передаватися безпосередньо від покупця до постачальника без помилок і затримки, які трапляються при трансляції замовлень покупців. Зміни в замовленні покупця можуть приводити до автоматичних змін у замовленнях постачальникам, зменшуючи кількість повторної роботи і затримки. Якість продуктів і коректність оформлення замовлення значно покращуються, а також зменшується час їх доставки.

Використання CSRP забезпечує постачальників інформацією про виробничі графіки замовників, даними про їх продаж, дозволяючи на підставі цієї інформації наперед планувати власне виробництво й поставки.

Запровадження інформаційних технологій покращує використання всіх ресурсів організації, підвищує її гнучкість і адаптованість до змін зовнішньої кон'юнктури, підвищує якість управлінських рішень і забезпечує відповідну конкурентоспроможність.

## Література

1. Вендров А.М. Один из подходов к выбору средств проектирования баз данных и приложений. "СУБД", 1995, №3.
2. Зиндер Е.З. Бизнес-реинжиниринг и технологии системного проектирования. Учебное пособие. М., Центр Информационных Технологий, 1996
3. Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). М., "Лори", 1996.
4. Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования. М., "МетаТехнология", 1993.
5. Международные стандарты, поддерживающие жизненный цикл программных средств. М., МП "Экономика", 1996
6. Создание информационной системы предприятия. "Computer Direct", 1996, N2
7. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях. Киев, "Диалектика", 1993.
8. Barker R. CASE\*Method. Entity-Relationship Modelling. Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co., 1990.
  1. Barker R. CASE\*Method. Function and Process Modelling. Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co., 1990.
  2. Boehm B.W. A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes, Aug. 1986
  3. Chris Gane, Trish Sarson. Structured System Analysis. Prentice-Hall, 1979.
  4. Edward Yourdon. Modern Structured Analysis. Prentice-Hall, 1989.
  5. Tom DeMarco. Structured Analysis and System Specification. Yourdon Press, New York, 1978.
  6. Westmount I-CASE User Manual. Westmount Technology B.V., Netherlands, 1994.
  7. Uniface V6.1 Designers' Guide. Uniface B.V., Netherlands, 1994.
  8. IEEE Std 1348-1995. IEEE Recommended Practice for the Adoption of CASE Tools.
  9. IEEE Std 1209-1992. IEEE Recommended Practice for the Evaluation and Selection of CASE Tools.
  10. PVCS Version Manager. User's Guide.
  11. PVCS Tracker. User's Guide.
  12. QA Partner. User's Guide.
13. Новоженев Ю.В. Объектно-ориентированные технологии разработки сложных программных систем. М., 1996.
14. Панащук С.А. Разработка информационных систем с использованием CASE-системы Silverrun. "СУБД", 1995, №3.
15. Горчинская О.Ю. Designer/2000 - новое поколение CASE-продуктов фирмы ORACLE. "СУБД", 1995, №3.
16. Горин С.В., Тандоев А.Ю. Применение CASE-засобіва Erwin 2.0 для информационного моделирования в системах обработки данных. "СУБД",

1995, №3.

17. Горин С.В., Тандоев А.Ю. CASE-засобіво S-Designor 4.2 для разработки структуры базы данных. "СУБД", 1996, №1.

18. DATARUN Concepts. Computer Systems Advisers Research Ltd., 1994.

19. SE Companion Installation and Administration Manual. SECA Inc., 1995.

20. Петров Ю.К. JAM - инструментальное средство разработки приложений в информационных системах архитектуры "клиент/сервер", построенных на базе РСУБД. "СУБД", 1995, №3.