

2.2. МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ ІНФОРМАЦІЙНИХ СИСТЕМ ПІДПРИЄМСТВ ТА ЙОГО ОСНОВНІ ЕТАПИ

Опис життєвого циклу інформаційної системи передбачає оперування такими поняттями:

- *процес* — ланцюжок робіт, що послідовно виконуються;
- *етапи* — послідовні відрізки часу, упродовж якого виконуються роботи. Протягом етапу можуть виконуватися роботи, що належать до різних процесів. В основі діяльності із створення й використання автоматизованої інформаційної системи управління підприємством (АІСУП) лежить поняття її *життєвого циклу* (ЖЦ). ЖЦ є моделлю створення і використання АІСУП, що відображає різні її стани, починаючи з моменту виникнення необхідності в даному виробі і закінчуючи моментом його повного виходу з використання всіх, без винятку, користувачів.

Традиційно виділяють такі **основні етапи ЖЦ АІСУП**:

- *аналіз вимог*;
- *проектування*;
- *програмування / впровадження*;
- *тестування і налагодження*;
- *експлуатація і супровід*.

ЖЦ утворюється відповідно до принципу низхідного проектування і зазвичай має ітераційний характер: реалізовані етапи, починаючи з найперших, циклічно повторюються відповідно до змін вимог і зовнішніх умов, введення обмежень тощо. На кожному етапі ЖЦ породжується певний набір документів і технічних рішень, при цьому для кожного етапу початковими є документи і рішення, отримані на попередньому етапі. Кожний етап завершується верифікацією породжених документів і рішень з метою перевірки відповідності їх вихідним.

Існуючі моделі ЖЦ визначають порядок виконання етапів у ході розробки, а також критерії переходу від етапу до етапу. Відповідно до цього найбільше поширення отримали такі три моделі ЖЦ:

1. Каскадна модель (70—80-ті роки) передбачає перехід до наступного етапу після повного завершення робіт на попередньому етапі і характеризується чітким поділом даних і процесів їх опрацювання (рис. 2.4).

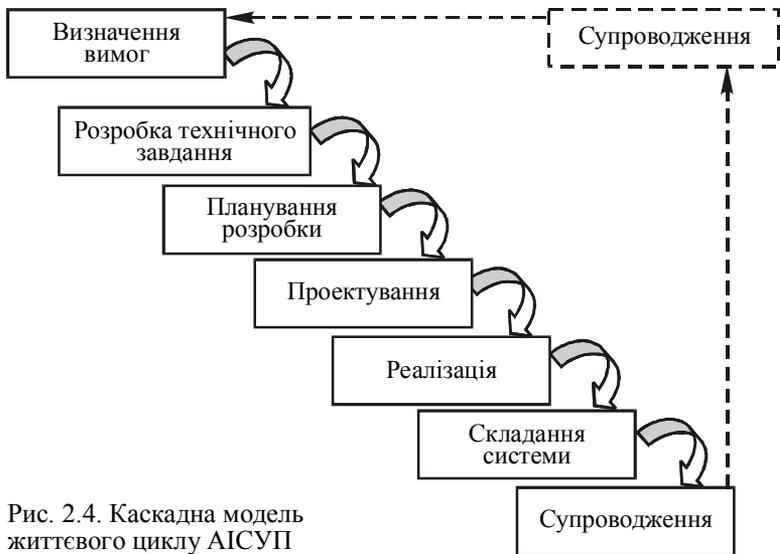


Рис. 2.4. Каскадна модель життєвого циклу АІСУП

2. Поетапна модель з проміжним контролем (80—85-ті роки) — ітераційна модель розробки з циклами зворотного зв'язку між етапами. Перевага такої моделі — в тому, що міжетапні коригування забезпечують меншу трудомісткість порівняно з каскадною моделлю; з іншого боку, час життя кожного з етапів розтягується на весь період розробки.

3. Спіральна модель (86—90-ті роки) — загострює увагу на початкових етапах ЖЦ: аналізі вимог, проектуванні специфікацій, попередньому й детальному проектуванні. На цих етапах

перевіряється і обґрунтовується реалізованість технічних рішень створенням прототипів. Кожний виток спіралі відповідає поетапній моделі створення фрагмента або версії системи, на ньому уточнюються цілі й характеристики проекту, визначається його якість, плануються роботи наступного витка спіралі. Таким чином поглиблюються і послідовно конкретизуються деталі проекту і в результаті обирається обґрунтований варіант, який доводиться до реалізації (рис. 2.5).

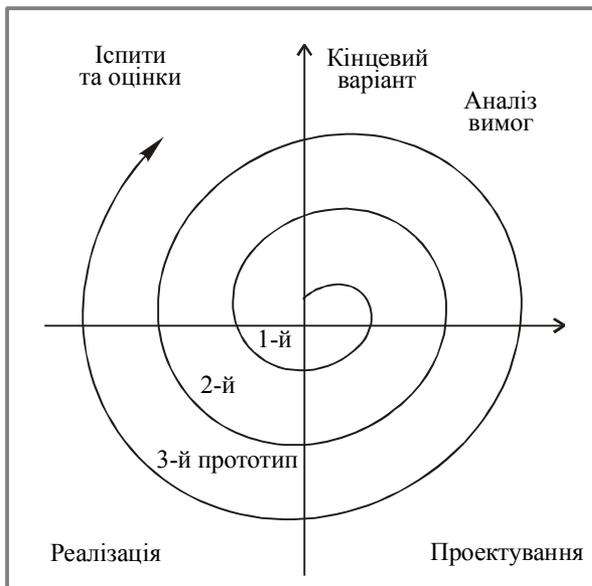


Рис. 2.5. Спіральна модель життєвого циклу АІСУП

Фахівці відзначають такі переваги спіральної моделі:

- накопичення і повторне використання програмних засобів, моделей і прототипів;
- орієнтація на розвиток і модифікацію системи в ході її проектування;
- аналіз ризику і витрат у процесі проектування.

Розгляньмо детальніше основні етапи ЖЦ.

1) Аналіз вимог.

Аналіз вимог є першою фазою розробки АІСУП, на якій вимоги замовника уточнюються, формалізуються і документуються. Фактично на цьому етапі дається відповідь на питання: «Що повинна робити майбутня система?» Саме тут лежить ключ до успіху всього проекту. У практиці створення великих систем відомо чимало прикладів невдалої реалізації проекту саме через неповноту і нечіткість визначення системних вимог.

Перелік вимог до АІСУП повинен включати:

- **сукупність умов**, за яких передбачається експлуатувати майбутню систему (апаратні й програмні ресурси, що надаються

системі; зовнішні умови її функціонування; склад працівників і робіт, що мають до неї відношення);

- **опис функцій**, що їх має виконувати система;
- **обмеження в процесі розробки** (директивні терміни завершення окремих етапів, наявні ресурси, організаційні процедури і заходи, що забезпечують захист інформації).

Метою аналізу є перетворення загальних, нечітких знань про вимоги до майбутньої системи в точні (по можливості) визначення. Результатом етапу повинна бути модель вимог до системи (іншими словами — системний проект), що визначає:

- **архітектуру системи, її функції, зовнішні умови, поділ функцій між апаратною і програмною частинами (ПЧ);**
- **інтерфейси і поділ функцій між людиною і системою;**
- **вимоги до програмних та інформаційних компонентів ПЧ, необхідні апаратні ресурси, вимоги до бази даних, фізичні характеристики компонент ПЧ, їхні інтерфейси.**

Модель вимог повинна включати:

- *повну функціональну модель вимог до майбутньої системи з глибиною опрацювання до рівня кожної операції кожної посадової особи;*

- *специфікації операцій нижнього рівня;*
- *пакет звітів і документів по функціональній моделі, що включає характеристику об'єкта моделювання, перелік підсистем, вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами, вимоги до характеристик взаємозв'язків системи із суміжними системами, вимоги до функцій системи;*

- *концептуальну інформаційну модель вимог;*
- *пакет звітів і документів з інформаційної моделі;*
- *архітектуру системи з прив'язкою до концептуальної інформаційної моделі;*

- *пропозиції щодо організації структури для підтримки системи.*

Таким чином, модель вимог містить функціональну, інформаційну і, можливо, подійну (у разі, якщо цільова система є системою реального часу) моделі, що забезпечує ряд переваг порівняно з традиційною моделлю:

1. Для традиційної розробки характерним є здійснення початкових етапів кустарними неформалізованими способами. Тому замовники і користувачі уперше можуть побачити систему після того, як вона вже значною мірою реалізована. Природно, ця система відрізнятиметься від тієї, якої вони очікували. Тож далі

матимуть місце ще декілька ітерацій її розробки або модифікації, що вимагає додаткових (і значних) витрат грошей і часу. Ключ до розв'язання цієї проблеми і дає модель вимог, що дозволяє:

- описати, «побачити» і скоригувати майбутню систему до того, як вона буде реалізована фізично;
- зменшити витрати на розробку і впровадження системи;
- оцінити розробку за часом і результатами;
- досягнути взаєморозуміння між усіма учасниками роботи (замовниками, користувачами, розробниками, програмістами);
- поліпшити якість системи, що розробляється, а саме: виконати її функціональну декомпозицію і спроектувати оптимальну структуру інтегрованої бази даних.

2. Модель вимог повністю незалежна і відокремлена від конкретних розробників, не вимагає супроводження її творцями і може бути безболісно передана іншим особам. Понад те, якщо з яких-небудь причин підприємство не готове до реалізації системи на основі моделі вимог, вона може бути залишена «на полиці» доти, доки в ній не виникне потреба.

3. Модель вимог може бути використана для самостійної розробки або коригування вже реалізованих на її основі програмних засобів силами програмістів відділу автоматизації підприємства.

4. Модель вимог може використовуватися для автоматизованого і швидкого навчання нових працівників конкретного напрямку діяльності підприємства, оскільки її технологія міститься в моделі.

Етап аналізу вимог є найважливішим серед усіх етапів ЖЦ. Він істотно впливає на всі подальші етапи, залишаючись водночас найменш вивченим і зрозумілим процесом. На цьому етапі, по-перше, потрібно зрозуміти, що саме треба зробити, а по-друге, задокументувати це, бо якщо вимоги не зафіксовані і не зроблені доступними для учасників проекту, то вони начебто й не існують. При цьому мова, якою формулюються вимоги, повинна бути досить простою і зрозумілою замовникові.

2) Розробка технічного завдання.

Після побудови моделі, що містить вимоги до майбутньої системи, на її основі розробляється **технічне завдання** зі створення системи, що включає в себе:

- *вимоги до автоматизованих робочих місць, їхніх складу і структури, а також до способів і схем інформаційної взаємодії між ними;*
- *розробку вимог до технічних засобів;*

- визначення вимог до програмних засобів;
- розробку топології, складу і структури локальної обчислювальної мережі;
- вимоги до етапів і термінів виконання робіт.

3) Проектування.

Етап проектування дає відповідь на питання: «Як (яким чином) система задовольнятиме вимоги, що ставляться до неї?». Завданням цього етапу є дослідження структури системи і логічних взаємозв'язків її елементів, причому тут не зачіпаються питання, пов'язані з реалізацією на конкретній платформі. Проектування розглядається як ітераційний процес отримання логічної моделі системи разом зі строго сформульованими цілями, поставленими перед нею, а також написання специфікацій фізичної системи, що задовольняє ці вимоги. Цей етап звичайно поділяють на два підетапи:

а) **проектування архітектури системи**, що включає розробку структури та інтерфейсів компонентів, узгодження функцій і технічних вимог до компонентів, методів і стандартів проектування;

б) **детальне проектування**, яке передбачає розробку специфікацій кожного компонента, інтерфейсів між компонентами, розробку вимог до тестів і плану інтеграції компонентів.

Іншими словами, проектування є етапом ЖЦ, на якому визначається, як слід реалізовувати вимоги до АІСУП, що породжені й зафіксовані на етапі аналізу. В результаті повинна бути побудована модель реалізації, що демонструє, як система задовольнятиме пред'явлені до неї вимоги (без технічних подробиць). Фактично модель реалізації є розвитком і уточненням моделі вимог, а саме проектування є мостом між аналізом і реалізацією.

4) Реалізація (програмування / адаптація).

На етапі реалізації здійснюється створення системи як комплексу програмно-апаратних засобів, починаючи з проектування і створення телекомунікаційної інфраструктури і завершуючи розробкою та інсталяцією додатків. Зараз існує обширна література, в якій досить докладно розглянуті всі ці процеси, включаючи сучасні методи генерації коду прикладних систем, що використовуються. Тому в цьому підручникові питання реалізації не розглядається.

5) Тестування і налагодження.

Коректність АІСУП є її найважливішою властивістю і, без сумніву, головним предметом турботи розробників. У ідеальному випадку під коректністю АІСУП мають на увазі відсутність у ній помилок. Однак для більшості складних програмних продуктів досягти цього неможливо — «у кожній програмі міститься принаймні одна помилка». Тому під «коректним» зазвичай розуміють програмний продукт, що працює відповідно до пред'явлених до нього вимог, іншими словами — продукт, для якого поки ще не знайдені такі умови, в яких він виявиться непрацездатним.

Встановлення коректності є головною метою етапу життєвого циклу, що розглядається. Треба зазначити, що етап тестування і налагодження — один із найбільш трудомістких, стомлюючих і непередбачуваних етапів розробки АІСУП. У середньому за розробки традиційними методами цей етап займає від 1/2 до 1/3 всього часу розробки. З іншого боку, тестування і налагодження являють собою серйозну проблему: у деяких випадках тестування і налагодження програми вимагають в декілька разів більше часу, ніж безпосередньо програмування.

Тестування — це набір процедур і дій, призначених для демонстрації коректної роботи АІСУП у заданих режимах і зовнішніх умовах. Мета тестування — виявити наявність помилок або переконливо продемонструвати їх відсутність, що можливо лише в окремих тривіальних випадках. Важливо розрізняти тестування і супутнє поняття «налагодження». Налагодження — це набір процедур і дій, що починаються з виявлення самого факту наявності помилки і закінчуються встановленням точного місця, характеру цієї помилки і способів її усунення.

Найважливішим і найчастіше застосовуваним на практиці є метод детермінованого тестування. При цьому як еталони тестів використовуються конкретні початкові дані, що складаються з взаємопов'язаних вхідних і результуючих величин і правильних послідовностей їх опрацювання. У процесі тестування із заданими початковими величинами треба встановити відповідність результатів їх опрацювання еталонним величинам. Для складних систем потрібна велика кількість тестів, і виникає проблема оцінки їх необхідної кількості й використання методів їх скорочення. Тому тестування (як і будь-який інший вид діяльності) доцільно планувати. План тестування повинен містити:

- *формулювання цілей тестування;*

- критерії якості тестування, що дозволяють оцінити його результати;
- стратегію проведення тестування, що забезпечує досягнення заданих критеріїв якості;
- потреби в ресурсах для досягнення заданого критерію якості за обраною стратегією.

● **Автоматизація тестування і налагодження**

Система автоматизації тестування і налагодження (САТН) являє собою складний комплекс алгоритмічних і програмних засобів, призначених для автоматизації аналізу АІСУП, тестування, налагодження й оцінок її якості, і дозволяє полегшити модифікацію компонент АІСУП, забезпечити виявлення помилок на ранніх стадіях налагодження, підвищити процент помилок, що автоматично виявляються.

б) Експлуатація і супроводження.

Основними завданнями етапу експлуатації і супроводження є такі:

- забезпечення стійкості роботи системи і збереження інформації — адміністрування;
- своєчасна модернізація і ремонт окремих елементів — технічна підтримка;
- адаптація можливостей системи, що експлуатується, до поточних потреб бізнесу підприємства — розвиток системи.

Ці роботи необхідно включати в оперативний план інформатизації підприємства, який повинен формуватися обов'язково з дотриманням усіх умов стратегічного плану. В іншому випадку в межах існуючої системи можуть з'явитися фрагменти, які в майбутньому зроблять ефективну експлуатацію системи неможливою. Зараз за рубежом стало загальноприйнятим передавати функції технічної підтримки і частково адміністрування постачальникам системи або системним інтеграторам. Ця практика одержала назву «аутсорсинг». Часто в межах аутсорсингу стороннім підприємствам передаються й такі функції, як створення і підтримка резервних сховищ даних і центрів виконання критичних бізнес-додатків, які задіюються у разі стихійного лиха або інших особливих умов.

Особливу увагу на етапі експлуатації і супроводу потрібно приділити питанням навчання персоналу і, відповідно, плануванню інвестицій у цей процес.