## ЛЕКЦІЯ 3. ЖИТТЄВИЙ ЦИКЛ ПРОЕКТУ

План лекції:

1. Поняття життєвого циклу програмного проекту.
2. Зв’язок моделі життєвого циклу програмного проекту і підходів до управління та показників програмних проектів.
3. Неструктурований життєвий цикл програмних проектів.
4. Каскадна модель життєвого циклу програмних проектів.
5. V-подібна модель життєвого циклу.
6. Модель життєвого циклу на основі розробки прототипів.
7. Інкрементна модель життєвого циклу програмного проекту.
8. Спіральна модель проектування.
9. Ітеративна модель життєвого циклу програмного проекту.

Питання, що виносяться на самостійне вивчення студентом (5 год.):

1. Об’єктно-орієнтована модель життєвого циклу [7, С.121-178].
2. Моделі швидкої розробки [2, С. 23-34].
3. Гібридні моделі життєвого циклу програмних проектів [7, С.164-169].

### *3.1.Поняття життєвого циклу програмного проекту*

Важливою рисою будь-якого проекту є процес його виконання.

Процес передбачає розтягнутість у часі і має також назву життєвий цикл, що більш точно відповідає таким обов’язковим складовим проекту як його дата початку і кінця.

Як уже зазначалося раніше, результатом проекту є продукт, який характеризується унікальністю. Унікальність продукту передбачає, що процес його створення також має бути в тій чи іншій мірі унікальним.

Однак менеджер проекту не може створювати кожного разу життєвий цикл «з нуля», замість цього він звертається до перевіреної і узагальненої практики, яка адаптується для конкретного проекту.

Існує багато моделей життєвого циклу, які з'являлися і розвивалися поступово. Менеджер проекту має розуміти переваги і недоліки кожної з них, для того, щоб обрати і використати правильну модель для кожного конкретного проекту.

### *3.2. Зв'язок моделі життєвого циклу програмного проекту і підходів до управління та показників програмних проектів*

Згідно з результатами аналізу Standish Group, проведеного за період з 1994 р. по 2000 р. на основі вивчення близько 30 тис. проектів з розробки ПЗ у різних країнах світу, значна кількість проектів з розробки ПЗ зазнала повної невдачі чи призвела до суттєвих перевитрат фінансових ресурсів і часу (рис. 1.5).

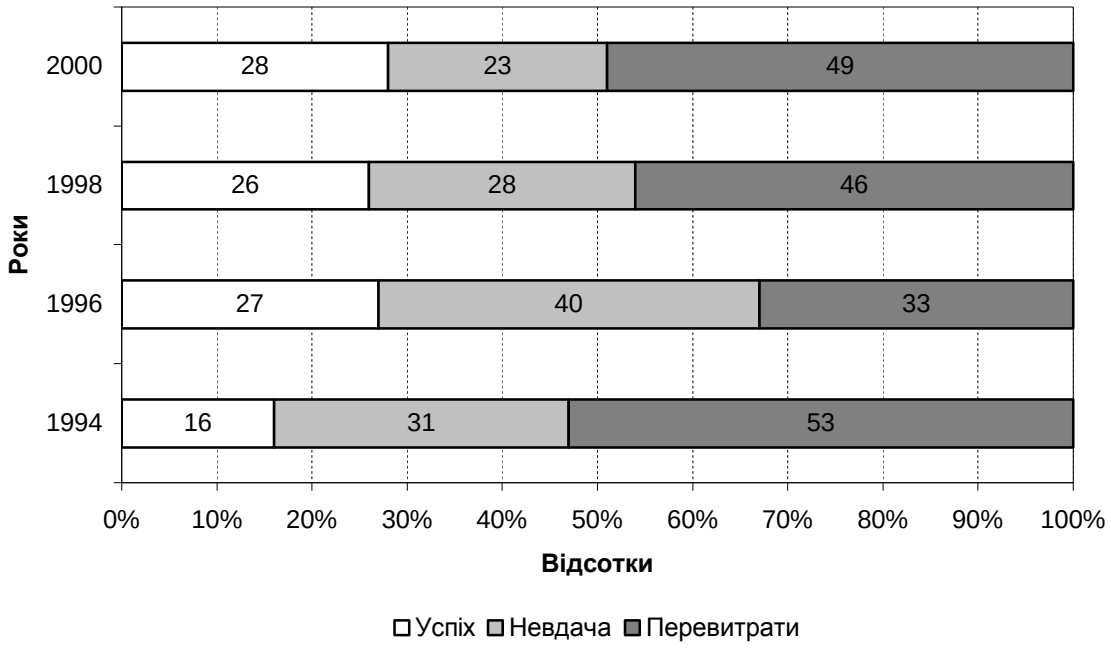


Рис. 3.1. Розподіл між проектами, що закінчилися успішно, невдачею чи призвели до суттєвих перевитрат фінансових ресурсів і часу за період 1994-2000 рр.

Незважаючи на тенденцію зростання кількості успішних проектів і скорочення кількості проектів, що закінчилися невдачею у 2000 р. в порівнянні з 1994 р., в цілому ситуація є вкрай негативною, оскільки

повністю успішними залишається значно менше третини проектів. Серед основних причин, згідно з результатами дослідження Standish Group, називається невміле чи неефективне управління проектами з розробки ПЗ.

Ефективне управління процесом розробки ПЗ можливе за умов повного розуміння процесів, що відбуваються в процесі роботи над проектом.

Життєвий цикл проекту представляє собою модель виконання проекту, від рівня відповідності якої підходам, що використовуються для управління, у значній мірі залежить кінцевий успіх проекту.

### *3.3.Неструктурований життєвий цикл програмних проектів*

На початкових етапах розробки ПЗ процес не був структурованим, його модель можна було представити як постановку завдання та його виконання до того часу, доки воно не буде зроблено, чи відмінено.

Можливість використання такої моделі відповідала характеру ПЗ для якого вона використовувалася – це, насамперед, ПЗ, що створювалося для наукових розрахунків. Подібне ПЗ не відрізнялося структурною складністю, було порівняно невеликим за обсягами.

Однак використання неструктурованої моделі життєвого циклу для великих і складних за структурою програмних проектів ускладнено, оскільки модель не відповідає процесам, що необхідні для створення подібного ПЗ.

### *3.4.Каскадна модель життєвого циклу програмних проектів*

Як було зазначено раніше, першою загальновизнаною моделлю життєвого циклу програмного проекту вважається модель SLC. Однак незворотній характер етапів моделі не відображав прийняту у практиці більшості компаній-розробників ПЗ можливість повернення на попередні етапи.

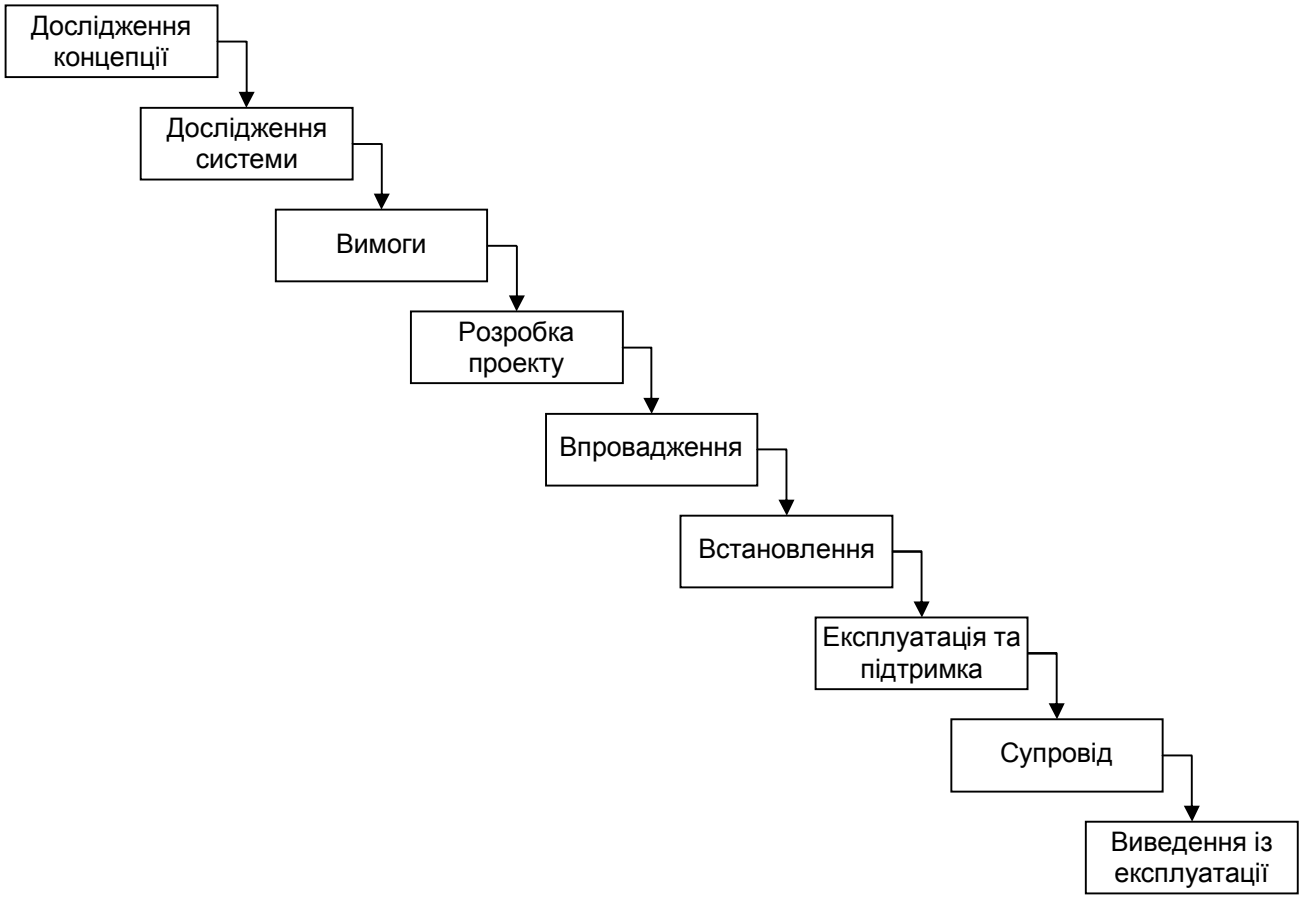


Рис. 3.2. Каскадна модель життєвого циклу розробки ПЗ SLC

Як уже зазначалося, в даний час модель SLC вважається застарілою і у «чистому» вигляді майже не застосовується, однак вона стала основою практично всіх сучасних моделей вивчення життєвого циклу розробки ПЗ, термінологія та послідовність етапів SLC були запозичені більш досконалими моделями, тому їх сутність та значення для характеристики діяльності з розробки ПЗ не втратили своєї актуальності в наш час.

### *3.5. V-подібна модель життєвого циклу*

На заміну каскадній моделі була запропонована V-подібна (шарнірна) модель, яка покликана усунути зазначені недоліки, зокрема поєднати етапи постановки завдання та перевірки його відповідності специфікаціям (рис. 3.4).

Однак «інженерний» підхід до розробки ПЗ, реалізований у моделі SLC та V-подібній моделі, виявився неконкурентноздатним, особливо для крупних проектів – тривала поетапна розробка проекту призводила до того, що процес розробки часто виходив за рамки встановленого бюджету та часових обмежень, а кінцевий продукт часто виявлявся таким, що запізнювався з виходом на ринок.

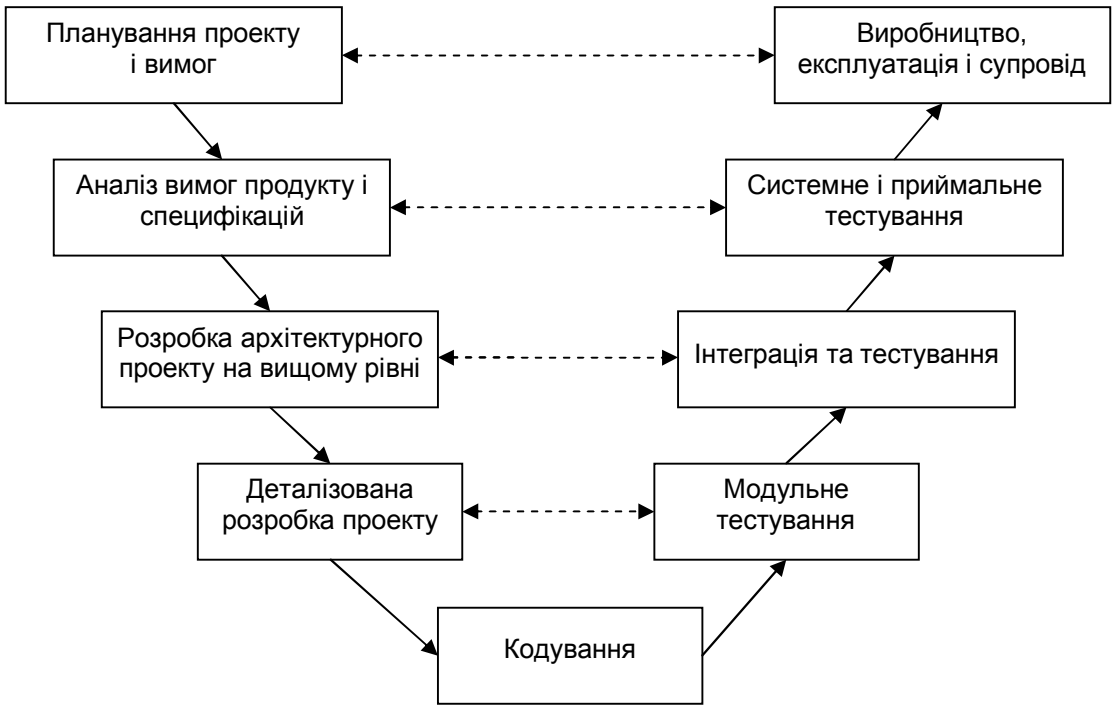


Рис. 3.3. V-подібна модель управління процесом розробки ПЗ

З погляду управління інноваціями, слід зазначити, що моделі SLC та V-подібна модель не пристосовані до можливості впровадження ефективного інноваційного менеджменту. Реалізація проекту може тривати декілька років, однак, відповідно до даних моделей, розроблені на його початку вимоги та план реалізації не враховують науково-технічний прогрес, вони орієнтовані на використання тих технологій, які були актуальними на початок робіт над проектом, що в результаті призводить до створення продукту який є морально застарілим та неконкурентноздатним ще до свого виходу на ринок.

### *3.6.Модель життєвого циклу на основі розробки прототипів*

Модель ЖЦ на основі розробки прототипів передбачає створення повністю або частково робочих моделей готової системи. Такий підхід схожий до використання зменшених моделей будівель у архітектурі.

Основний недолік моделі – наслідок конфлікту інтересів – можливість бути втягненим у нескінченний цикл вдосконалення продукту ще до його виходу у світ.

### *3.7.Інкрементна модель життєвого циклу програмного проекту*

Інкрементна модель ЖЦ передбачає розробку і поставку продукту частинами.

Перевага моделі – вона дозволяє рухатися вперед, зберігаючи інтереси обох сторін.

Недолік – зростання складності проекту та вартості і тривалості його розробки.

### *3.8.Спіральна модель проектування*

Вирішити основні недоліки моделі SLC покликана спіральна модель, яка відповідно до Б. Боема являє собою рухомий ризиком генератор процесної моделі, тобто являє собою модель процесу, в якій ризик виступає у якості фактору, що примушує її до розвитку.

Дана модель використовується для управління проектами з розробки програмного забезпечення, в яких бере участь значна кількість розробників. Вона має дві головні особливості: перша полягає в тому, що використовується циклічний підхід до поступового нарощування рівня визначеності та реалізації системи в умовах зниження її рівня ризику; друга являє собою набір певних віх (чи контрольних точок), котрі необхідні для того, щоб була можливість відслідковувати реальність впровадження та двостороннє узгодження рішень.

Ризик у даному випадку являє собою ситуації чи можливі випадки, що можуть стати на заваді проекту при досягненні ним певної мети (рис. 3.4).

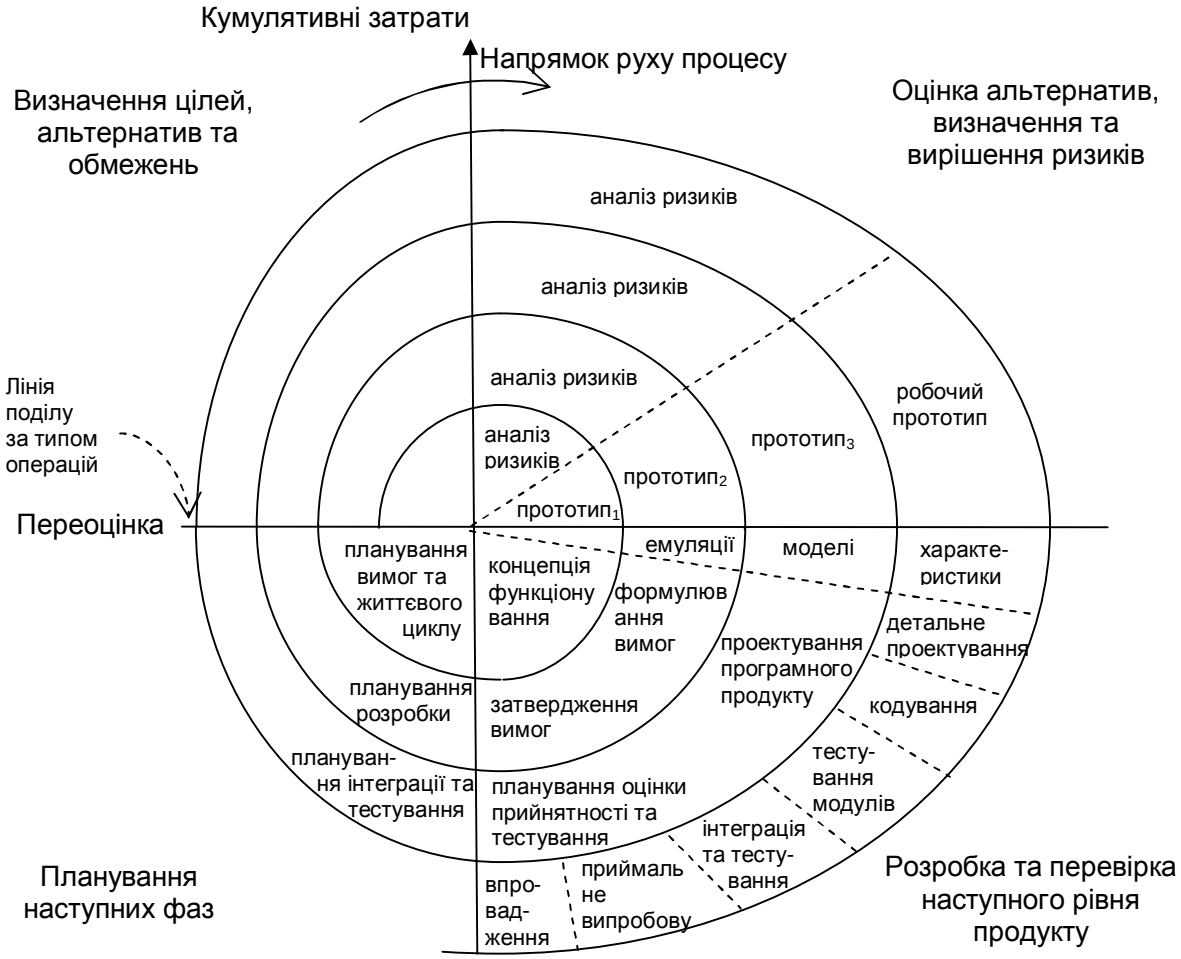


Рис. 3.4. Діаграма спірального розвитку

Відповідно до рис. 3.4 можна відзначити, що рух до кінцевої точки, який відбувається у напрямку руху годинникової стрілки на даній діаграмі, проходить значну кількість стадій, на кожній з яких, по-перше, відбувається уточнення функціональності кінцевого продукту, а, по-друге, спостерігається постійне зростання кумулятивних затрат на проект, які можна визначити у кожен конкретний момент часу роботи над проектом як абсолютну величину відстані між кожною точкою на спіралі та точкою перетину осей на діаграмі.

На відміну від попередніх розглянутих моделей, спіральна модель значно краще підходить до ефективного управління інноваціями. Окремі

витки спіралі розпочинаються з етапів оцінки ризику та планування, які можуть включати також і підбір та оцінку можливих інновацій у діяльності підприємства. Таким чином, кожен виток спіралі може передбачати інноваційну діяльність і розробка може бути спланована з урахуванням переваг, які будуть отримані в результаті впровадження інновацій у роботі підприємства-розробника.

Щоб підтвердити переваги спіральної моделі саме з погляду можливостей впровадження інновацій, Б. Боем наводить наступний приклад, згідно з яким на початку 1980-х рр. одна урядова установа стала замовником потужної інформаційної системи, що повинна була забезпечувати роботу понад однієї тисячі користувачів, розміщувалася у великому будівельному комплексі, мала потужні аналітичні та запитові можливості до великої динамічної бази даних. У контракті на розробку даної інформаційної системи була зафіксована вимога, згідно з якою час реакції системи на дії користувача не повинен перевищувати однієї секунди.

Використовуючи класичний підхід (каскадну модель) до проектування ПЗ і розробивши близько двох тисяч сторінок документів з вимогами, архітектори з програмного забезпечення визначили, що поставлене завдання можна реалізувати, побудувавши систему за Архітектурою А, приблизна вартість якої склала близько 100 млн. дол. США. Однак зазначена сума перевищувала рамки встановленого бюджету і реалізація проекту за таких умов була неможливою.

Лише завдяки створенню прототипу інтерфейсу користувача дослідним шляхом було встановлено, що вимога щодо часу реакції системи на дії користувача, який не повинен перевищувати однієї секунди, є завищеною, і достатньо затримки до чотирьох секунд. В результаті переробки вимог та вибору принципово іншої архітектури системи оціночна вартість системи була знижена до 30 млн. дол. США (рис. 3.5).

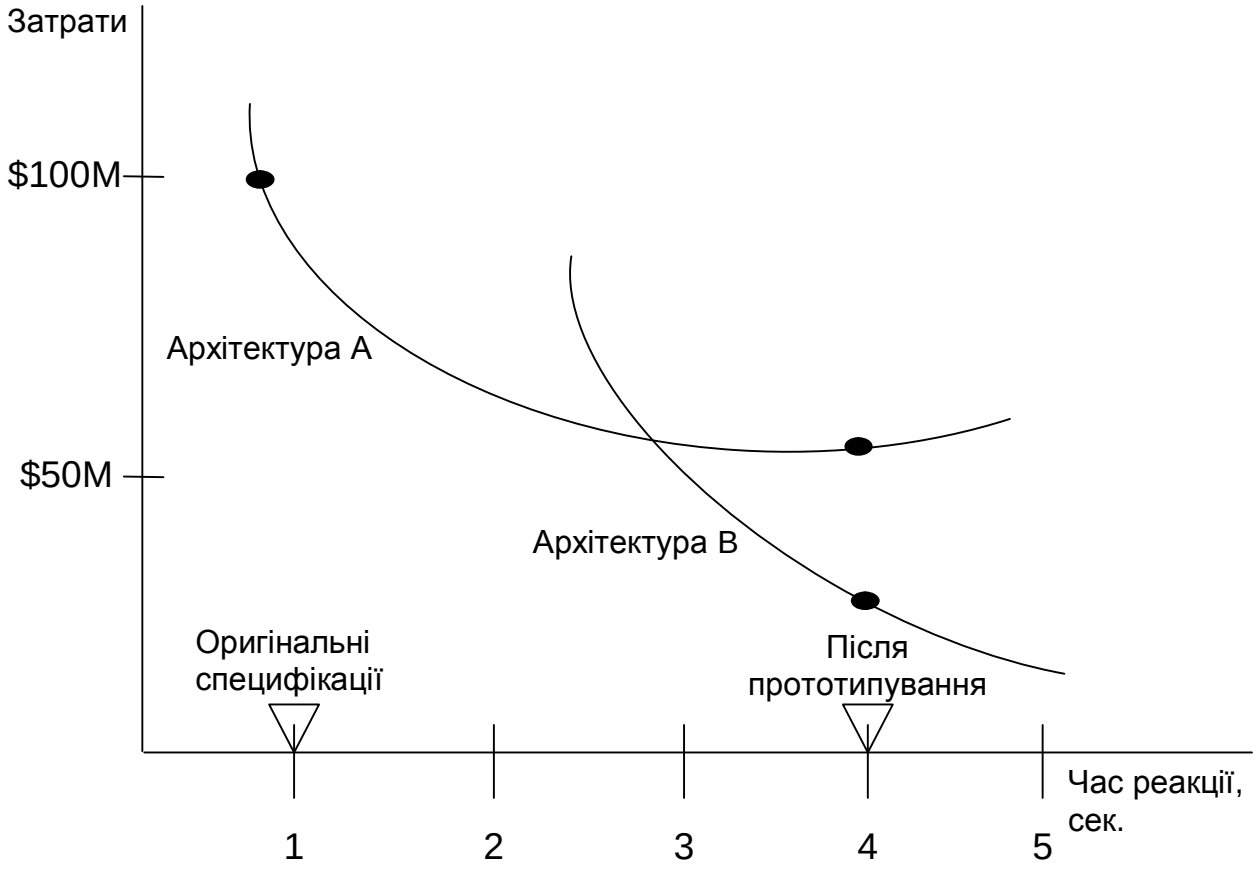


Рис. 3.5. Дві архітектури системи: затрати у порівнянні з часом реакції

На рис. 3.5 зображено дві можливі архітектури побудови системи. Кожна з них має свою сферу застосування: архітектура А здатна забезпечити більш високу швидкість реакції системи, проте архітектура В за умов невисоких вимог до швидкості реакції має значно нижчу вартість реалізації.

У тому випадку, коли при проектуванні використовується каскадна модель, можливі варіанти, коли після проектування за умов зміни вимог до системи рівень економії виявиться не достатньо високим, оскільки не розглядаються інші, можливо інноваційні, підходи (у даному випадку, коли використовується архітектура А і рух йде по верхній кривій на рис. 3.5 зліва направо).

Якщо ж за основу використовується спіральна модель, то прототип системи співвідноситься з ризиками. У даному випадку на початку кожного нового витку проектування системи визначається проблема й відшукуються її найбільш оптимальні шляхи вирішення: архітектура В буде обрана як

найбільш оптимальний варіант на одній з ітерацій проектування за умов ідентифікації ризику завищених вимог (часу реакції системи менше однієї секунди) та їх зміщення до 4 сек (рис. 3.5).

Таким чином, початок кожного нового витка спіралі може бути здійснений з переоцінки ризиків та визначення рішень і технологій, які доцільно застосувати у проекті. Саме з цієї точки зору спіральна модель значним чином відповідає основним положенням ефективного інноваційного менеджменту: початок кожного нового витка може бути здійснено з визначення нововведень, впровадження яких у процес розробки принесе найбільший економіко-технічний ефект.

### *3.9. Ітеративна модель життєвого циклу програмного проекту*

Подальший розвиток ідей спіральної моделі був здійснений у так званому «ітеративному підході», який вперше був запропонований Ф. Крачтеном.

Оригінальна спіральна модель Б. Боема використовує ітеративний підхід лише по відношенню до процесу проектування, інші складові процесу розробки ПЗ є послідовними, як у каскадній моделі. Ф. Крачтен поширив ітеративний підхід на усі інші складові процесу розробки ПЗ, виділивши при цьому чотири фази життєвого циклу:

1. Початок. На даній стадії команда приділяє основну увагу розумінню бізнес-варіанта проекту, визначенню його масштабу, можливості досягнення реалізації. Проводиться аналіз проблеми, складається документ-концепція, попередньо оцінюються графік, бюджет, а також фактори ризику проекту.
2. Дослідження. Уточнюються вимоги до системи, задається вихідна архітектура, демонструється попередній прототип системи.
3. Побудова. Головна увага приділяється реалізації. На цій стадії створюється основна частина програмного коду, закінчується планування та доробка архітектури.

4. Впровадження. Проводиться бета-тестування; користувачі та команда супроводження системи отримують досвід роботи з нею. Протестована базова версія ПЗ передається користувачам та розгортається для використання.

Ітеративна модель передбачає послідовний випуск життєздатних версій по закінченню кожної ітерації на усіх фазах життєвого циклу процесу розробки (рис. 3.6). В своїй основі ітеративна модель у значній мірі подібна спіральній, за тією відмінністю, що окремі ітерації (витки спіралі) охоплюють не лише процес проектування, а й усі інші процеси, для яких це можливо (процеси поділяються на робочі та допоміжні, кожен процес може мати свій цикл ітерацій, що не завжди має відповідати циклу інших процесів).

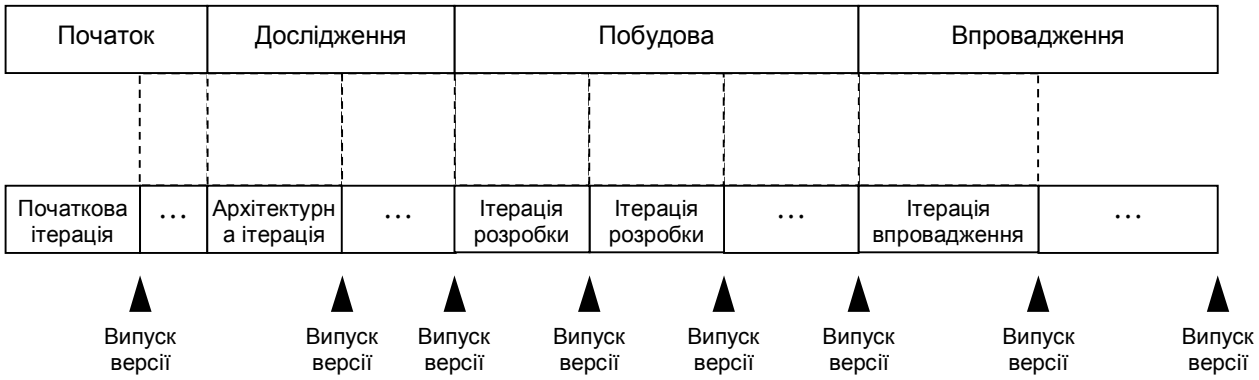


Рис. 3.6. Фази та ітерації, що призводять до появи життєздатних версій

Оскільки ітеративна модель розвинула основі положення спіральної моделі і, фактично, замінила її, то звичайно у сучасних умовах під спіральною моделлю процесу розробки ПЗ розуміють не оригінальну спіральну модель, розроблену Б. Боемом, а ітеративну модель Ф. Крачтена. Сам підхід набув широкої популярності і став відноситися до управління будь-якими проектами, а не лише проектами з розробки ПЗ [107].

У 1998 р. Б. Боем у співавторстві з іншими дослідниками запропонував модифікацію спіральної моделі, яка отримала назву спіральної моделі «Win-

Win». Спіральна модель «Win-Win» фактично є представленням гри двох учасників, причому умовою виграшу для них є успішне закінчення проекту.

На наш погляд, спіральна та ітеративна моделі, на відміну від попередніх, мають значні можливості для ефективного впровадження інновацій у діяльності з розробки ПЗ. Початок кожної нової ітерації доцільно розпочинати з процесу вибору та оцінки нововведень з максимальним економіко-технічним ефектом як для конкретного проекту, так і для фірми-розробника взагалі з урахуванням можливих ризиків, зокрема, затримки у реалізації проекту з причин впровадження інновацій.