УДК 004.453

***С.М. Козьменко**, д-р екон. наук, проф., **В.В. Колдовський**,*
Українська академія банківської справи Національного банку України

СУЧАСНІ МОДЕЛІ УПРАВЛІННЯ ПРОЦЕСОМ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Постановка проблеми. Управління процесом розробки програмного забезпечення (ПЗ) відрізняється особливою складністю та слабкою передбачуваністю кінцевого результату. Наявність різних методологій і підходів до розробки передбачають використання різних способів до моделювання даного процесу.

Невирішені раніше частини проблеми. Задача вибору відповідної моделі процесу розробки ПЗ залишається актуальною. Особливого значення вона набуває за умов зростання популярності слабоформалізованих підходів до розробки ПЗ, таких як “екстремальне програмування”.

Аналіз досліджень і публікацій. В основному моделі управління процесом розробки ПЗ проектувалися спеціалізованими установами, що розробляли стандарти розробки ПЗ і менеджменту програмних проектів.

Найбільш вагомі внески становлять дослідження, проведені фахівцями Інституту розробки програмного забезпечення (*Software Engineering Institute, SEI*), що є структурною одиницею Університету Карнегі-Меллона розташованого у США. Інститут працює у сфері дослідження питань менеджменту процесу розробки ПЗ, вивчає технічні аспекти розробки ПЗ та його готовність до подальшого розвитку. Інститут вперше у 1991 р. запропонував Модель характеристики зрілості процесу розробки ПЗ (*Software Capability Maturity Model, CMM*), котра набула загального визнання [3].

В Європі аналогом *SEI CMM* виступають стандарти серії *ISO*-9000 [5].

Вітчизняна практика не має усталеної загальноприйнятої системи організації роботи софтверних компаній та певної наукової школи, яка б пропагувала власний підхід до вирішення даних питань.

На рис. 1 представлені найбільш важливі стандарти та моделі управління процесом розробки ПЗ за період з 1985 по 2005 р.

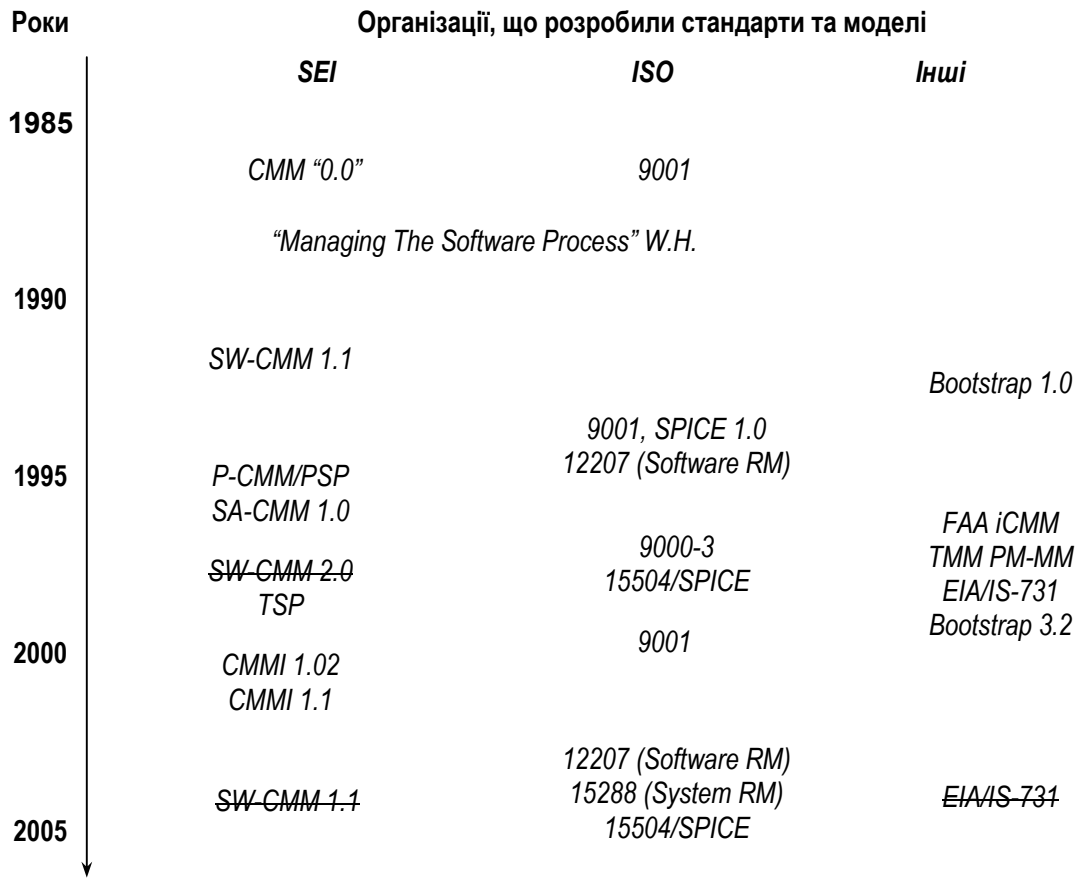

Рис. 1. Важливі стандарти і моделі управління розробкою програмного забезпечення з 1985 по 2005 р. [4]

Примітка: закресленими позначено ті стандарти, що втратили своє значення у сучасних умовах.

Позначення:

A – Acquisition (Злиття); *CMM – Capability Maturity Model* (Модель характеристики зрілості); *CMMI – CMM Integrated* (Інтегрована модель характеристики зрілості); *EIA*/IS – *Electronic Industries Alliance Interim Standard*; *PM-MM – Project Management Maturity Model* (Модель зрілості управління проектом); *PSP – Personal Software Process* (Персональний програмний процес); *RM – Reference Model* (Модель для посилання); *SE – Systems Engineering* (Системний інжиніринг); *Software* – Програмне забезпечення; *SPICE – Software Process Improvement Capability dEtermination* (Визначення можливостей для вдосконалення програмного процесу); *SW – Software Engineering* (Програмний інжиніринг); *TSP – Team Software Process* (Командний програмний процес).

Цілі статті. Дане дослідження покликане розкрити питання вибору відповідної моделі процесу управління, що відповідає прийнятій в організації методології розробки ПЗ.

Виклад основного матеріалу. Моделювання процесу розробки ПЗ почалося зі створення концепції життєвого циклу ПЗ (*SLC, Software Life Cycle*), що стала наслідком проведеної у жовтні 1968 р. конференції підкомітету НАТО з науки і техніки [1]. Модель *SLC* отримала назву "каскадної" чи "водоспадної", її графічне представлення *SLC* зображено на рис. 2.

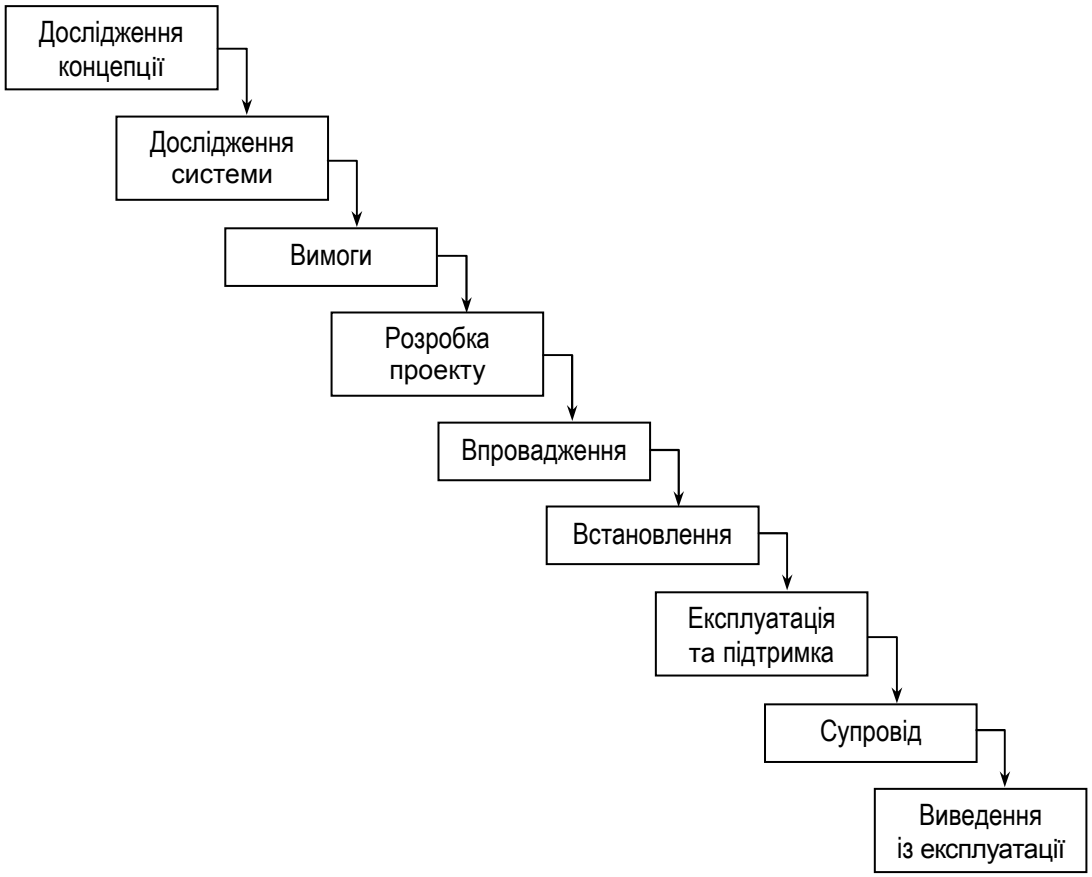

Рис. 2. Каскадна модель життєвого циклу розробки ПЗ *SLC* [1]

В даний час модель *SLC* вважається застарілою і у "чистому" вигляді майже не застосовується, оскільки процес розробки ПЗ у ній представлено аналогічно інженерній діяльності, що на практиці не відповідає дійсності.

Головний недолік моделі *SLC* – відсутність можливості повернення на попередній етап, що на практиці зустрічається дуже часто, наприклад, з метою уточнення специфікацій чи повернення на доробку після виявлених помилок у тестуванні. Саме тому модель *SLC* була модифікована з урахуванням зазначеної вимоги і у подальшому використовувалася у такому вигляді (рис. 3).

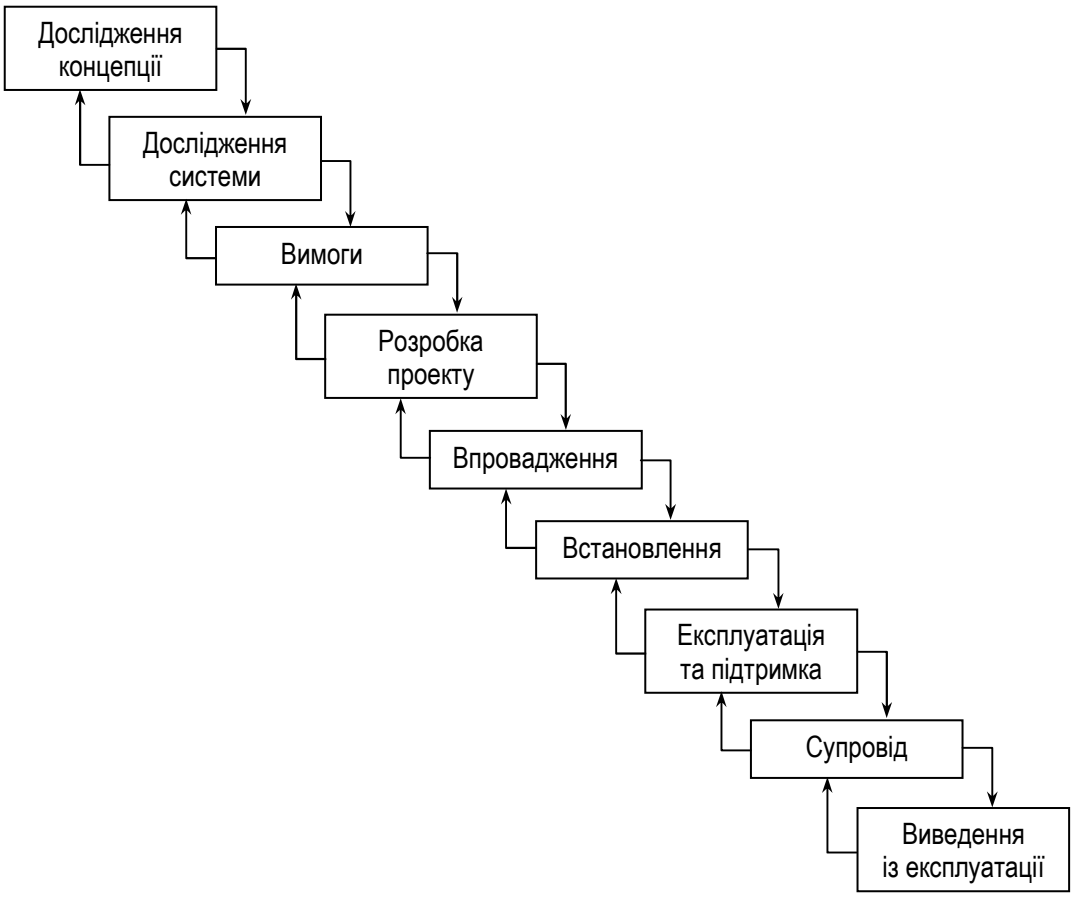

Рис. 3. Модифікована модель життєвого циклу розробки ПЗ *SLC* [1]

Однак навіть з урахуванням можливості повернення на попередній етап, модель *SLC* виявилася такою, що слабко відповідає вимогам до ефективного управління процесом розробки ПЗ.

Перехід з однієї фази *SLC* на іншу супроводжувався передачею проекту іншій команді учасників, виникали проблеми невідповідності вимог і специфікацій розробленому продукту, команди працювали розрізнено і з низькою ефективністю комунікацій.

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

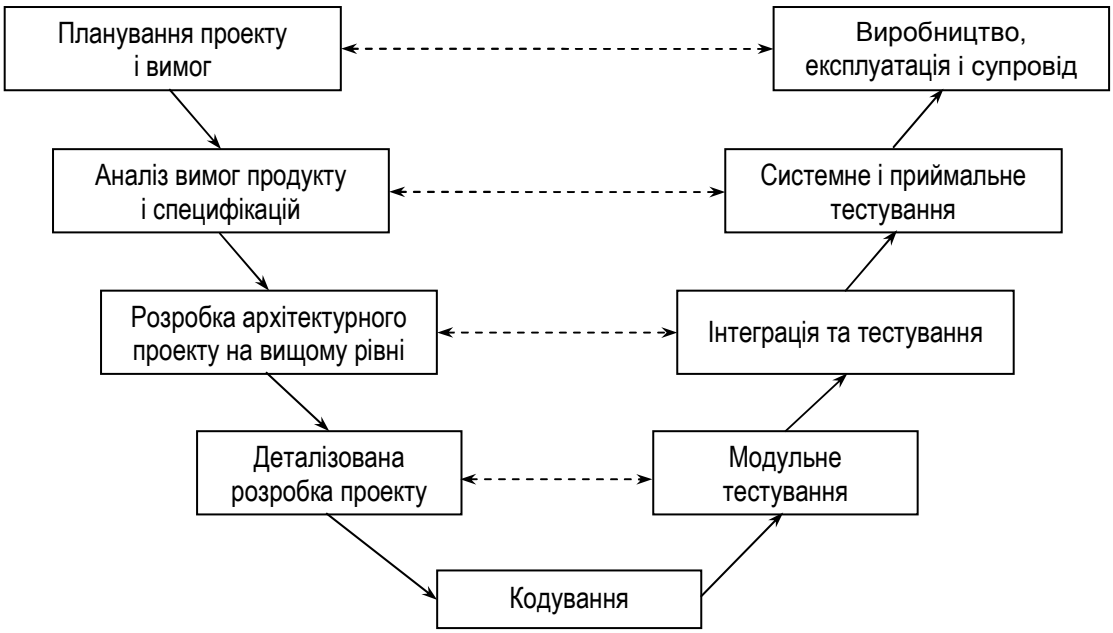

Рис. 4. V-подібна модель управління процесом розробки ПЗ

Вирішити основні недоліки моделі *SLC* покликана спіральна модель, яка у відповідності з роботою [2] являє собою рухомий ризиком генератор процесної моделі, тобто модель процесу, в якій ризик виступає як фактор, 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. 5) [2].

У відповідності з рис. 5 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 та розширення 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.

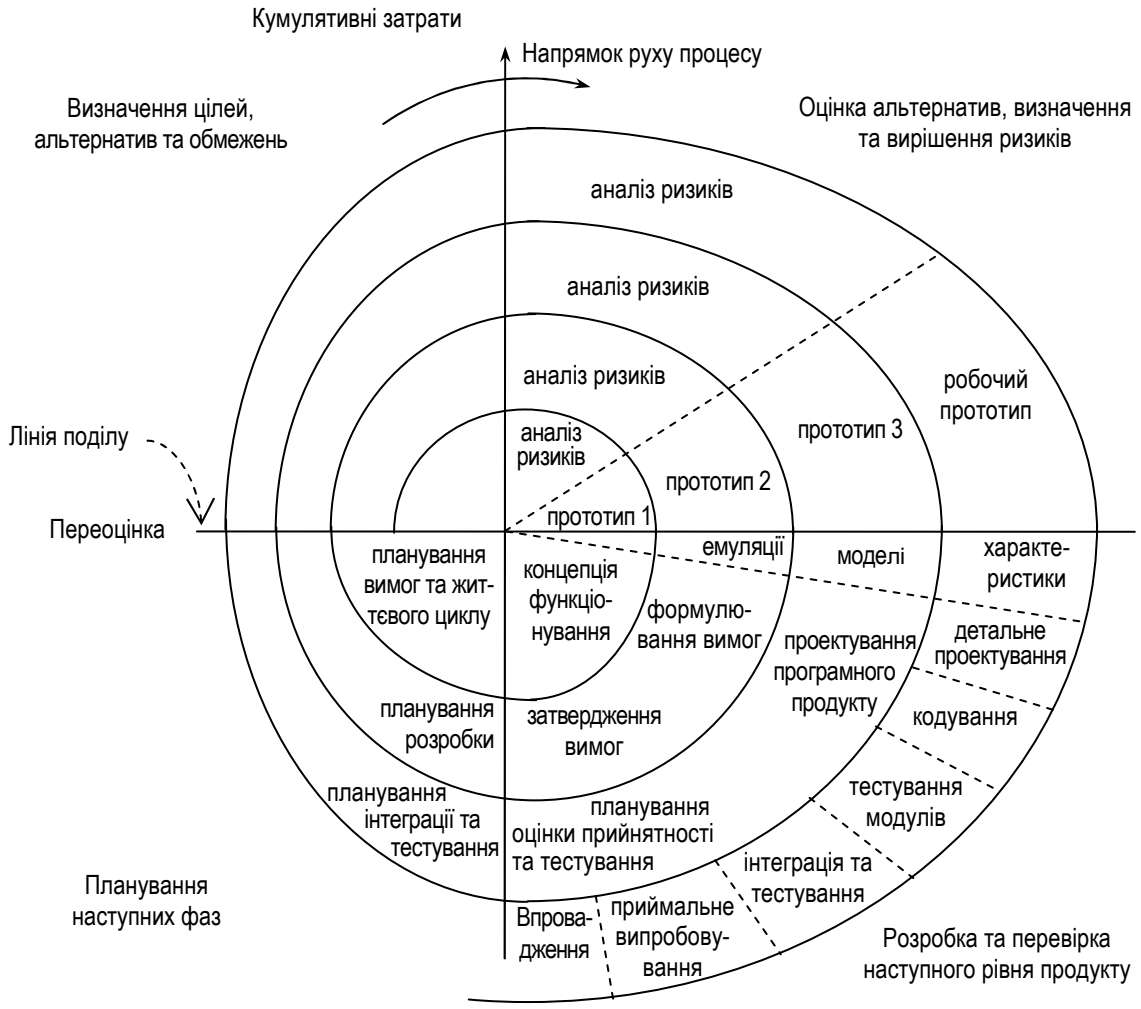

Рис. 5. Оригінальна діаграма Спірального розвитку [1]

В цілому модель Спірального розвитку зарекомендувала себе з позитивної сторони як така, що найбільше відповідає особливостям такого виду діяльності як розробка ПЗ. В даний час подібна модель використовується багатьма передовими компаніями галузі ПЗ, зокрема, *Microsoft*.

Висновки. Проблема вибору відповідної моделі управління процесом розробки ПЗ є актуальною в умовах глобалізації ринку ПЗ та суттєвого підвищення рівня конкуренції.

Модель управління процесом розробки має відповідати цілям і особливостям компанії-розробника. Перші моделі, такі як модель *SLC*, чи *V*-подібна модель, розглядали процес розробки ПЗ як аналог звичайної інженерно-конструкторської діяльності. Однак на практиці їх застосування виявилося недоцільним.

Найбільш поширеною в даний час є модель Спірального розвитку, яка дозволяє поступово скоригувати вимоги і функціональність проекту в процесі роботи над ним з урахуванням можливих ризиків.

Список літератури

1. Шафер Д., Фатрелл Р., Шафер Л. Управление программными проектами: достижение оптимального качества при минимуме затрат: Пер. с англ. – М.: “Вильямс”, 2003. – 1136 с.
2. Boehm B. Spiral Development: Experience, Principles, and Refinements / Spiral Development Workshop, Feb. 9, 2000 // Software Engineering Institute. – Pittsburgh (USA). – 2000. – 37 p.
3. Capability Maturity Model for Software, Version 1.1 / PaulkM., Curtis B., ChrissisM., Weber C. – Software Engineering Institute. – Pittsburgh (USA). – 1993. – 91 p.
4. Keefer G., Lubecka H. The CMMI in 45 minutes. Stuttgard: AVOCA GmbH. 2005. – 10 p.
5. Paulk M. A Comparison of ISO 9001 and the Capability Maturity Model for Software. – Software Engineering Institute. – Pittsburgh (USA). – 1993. – 78 p.

Козьменко, С.М. Сучасні моделі управління процесом розробки програмного забезпечення / С.М. Козьменко, В.В. Колдовський // Проблеми і перспективи розвитку банківської системи України: зб. наук. праць.- Суми: УАБС НБУ, 2005.- Т. 12.- С. 43-49.