

Лекція

Тема: Життєвий цикл ПЗ

1. Процеси циклу життя програмного забезпечення.
2. Типи моделей життєвого циклу ІТ-проектів.

1. Процеси циклу життя програмного забезпечення.

За десятки років розробки програмного забезпечення і програмних систем створено низку типових схем упорядкування виконання робіт з проектування і розроблення. Такі схеми одержали назву життєвого циклу і узагальнені в стандарті ISO/IEC 12207 і основоположних моделях ЖЦ, що застосовуються на практиці.

Характеристика життєвого циклу стандарту ISO/IEC 12207

Стандарт ISO/IEC 12207:2002 визначає загальну структуру і зміст ЖЦ ПС, починаючи з розробки концепції до утилізації системи. Структурно він складається з опису багатьох процесів, взаємозв'язків між ними, а також сформульованих дій і задач, виконуваних у цих процесах. Іншими словами, стандартний життєвий цикл визначає лише схему робіт за процесами розробки ПС, а не те, як саме виконувати ті або інші процеси (табл.2.1).

Таблиця 2.1. Процеси життєвого циклу в стандарті ISO/IEC 12207

№ п/п	Процес (підпроцес)
1. Категорія «Основні процеси»	
1.1	Замовлення (договір)
1.1.1	Підготовка замовлення, вибір постачальника
1.1.2	Моніторинг діяльності постачальника, приймання споживачем
1.2	Постачання (придбання)
1.3	Розроблення
1.3.1	Виявлення вимог
1.3.2	Аналіз вимог до системи
1.3.3	Проектування архітектури системи
1.3.4	Аналіз вимог до ПЗ системи
1.3.5	Проектування ПЗ
1.3.6	Конструювання (кодування) ПЗ
1.3.7	Інтеграція ПЗ
1.3.8	Тестування ПЗ
1.3.9	Системна інтеграція
1.3.10	Системне тестування
1.3.11	Інсталяція ПЗ
1.4	Експлуатація
1.4.1	Функціональне використання
1.4.2	Підтримка споживача
1.5	Супроводження
2. Категорія «Процеси підтримки»	
2.1	Документування
2.2	Керування конфігурацією
2.3	Забезпечення гарантії якості
2.4	Верифікація
2.5	Валідація
2.6	Загальний огляд

2.7	Аудит
2.8	Вирішення проблем
2.9	Забезпечення застосовності продукту

2.10	Оцінювання продукту
3. Категорія «Організаційні процеси»	
3.1	Керування
3.1.1	Керування на рівні організації
3.1.2	Керування проектом
3.1.3	Керування якістю
3.1.4	Керування ризиком
3.1.5	Організаційне забезпечення
3.1.6	Вимірювання
3.1.7	Керування знаннями
3.2	Удосконалення
3.2.1	Упровадження процесів
3.2.2	Оцінювання процесів
3.2.3	Удосконалення процесів

Стандарт не зобов'язує використовувати всі процеси ЖЦ одночасно і не ставить особливих вимог до формату і змісту розроблених документів. Тому організація-користувач стандарту під час розроблення конкретного програмного продукту може створити стандарти підприємства, методики і процедури, що деталізують вибрані для конкретних потреб процеси ЖЦ. Міжнародна організація зі стандартизації ISO (International Organization for Standardization) випускає також посібники і настанови, що доповнюють стандарт ISO/IEC 12207.

Як видно з табл. 2.1, усі процеси в даному стандарті поділяються на три категорії:

- основні процеси;
- процеси підтримки;
- організаційні процеси.

Для кожного з процесів визначені види діяльності (дії – activity), задачі, сукупність результатів (виходів) діяльності і розв'язання задач, а також деякі специфічні вимоги. У стандарті наведено перелік робіт для основних, організаційних процесів і процесів підтримки, але не спосіб їх виконання і не форма подання результатів.

В стандарті до основних процесів належать:

– *процес придбання*, який ініціює ЖЦ ПС і визначає дії організації-покупця (або замовника), що отримує автоматизовану систему, програмний продукт або сервіс. Цей процес містить у собі такі види діяльності: ініціювання і підготовка запиту, оформлення контракту і його актуалізація; моніторинг користувачів, приймання і завершення;

– *процес постачання*, який визначає дії з передачі покупцю програмного продукту або сервісу і містить у собі такі види діяльності: підготовку пропозицій (відповідей на запити); оформлення контракту; планування, виконання і контроль продукту, що постачається; аналіз і оцінку продукту; постачання і завершення робіт з постачання. Процес постачання починається тоді, коли встановлені договірні відношення між замовником і постачальником. Залежно від умов договору процес постачання може містити у собі процес розроблення ПЗ, процес експлуатації і супроводження для виправлення і поліпшення ПС;

– процес розроблення, який визначає дії підприємства-розробника програмного продукту: аналіз вимог до системи; проектування архітектури системи, детальне проектування компонентів ПС, кодування і тестування ПС, інтеграцію системи, кваліфікаційне тестування, установку ПС і забезпечення приймання ПС (рис.2.1);

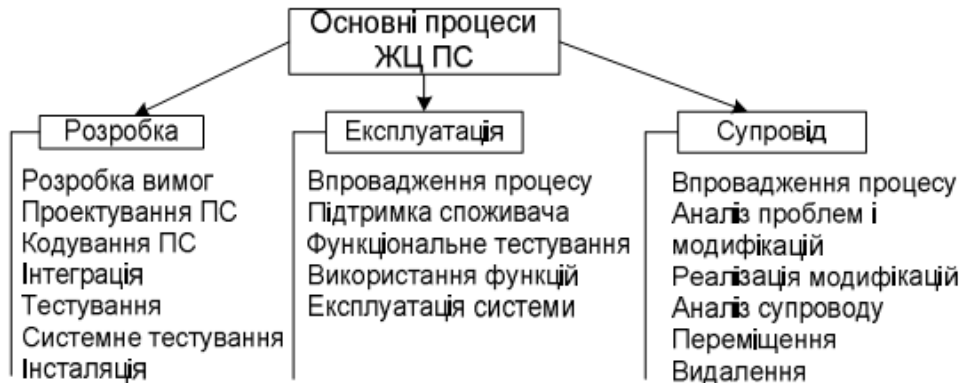


Рис. 2.1. Схема основних процесів ЖЦ ПС

– процес експлуатації, який визначає дії підприємства-оператора, що забезпечує обслуговування системи в ході її експлуатації користувачами (консультування користувачів, вивчення потреб операторів, задоволеності споживачів системою тощо). Цей процес регламентує задачі і дії з функціонального тестування, перевірки правильності експлуатації системи; дотримання інструкцій і настанов з її запуску;

– процес супроводження, який визначає дії організації, що виконує супровід програмного продукту (керування модифікаціями, підтримку поточного стану і функціональної придатності, інсталяцію програмного продукту на обчислювальній системі користувача та її вилучення при списанні). Даний процес містить у собі завдання і дії щодо аналізу питань супроводження і модифікації, розробки планів і реалізації модифікації системи, аналізу результатів супроводження після змін системи, міграції ПС в інше середовище або її виведення з експлуатації.

До категорії основних процесів належать також «первинні» процеси, що визначають порядок підготовки договору на розробку ПС, моніторинг діяльності постачальників ПС тощо (див. табл.2.1).

Стандарт містить у собі опис допоміжних процесів, що регламентують додаткові дії з перевірки продукту, керування проектом та його якістю (рис.2.2).

До процесів підтримки розроблення ПС належать: документування, керування версіями, верифікація і валідація, перегляди, аудити, оцінювання продукту та ін. Процес керування версіями за змістом відповідає керуванню конфігурацією системи, що так само, як і продукти процесів, повинні перевірятися на правильність реалізації цілей проекту і відповідність вимогам замовника. Завдання з перевірки рекомендується виконувати спеціальним контролерам, які знаються на методах і процесах проектування ПС.

До організаційних процесів належать: керування проектом (менеджмент розробки), якістю, ризиками тощо.

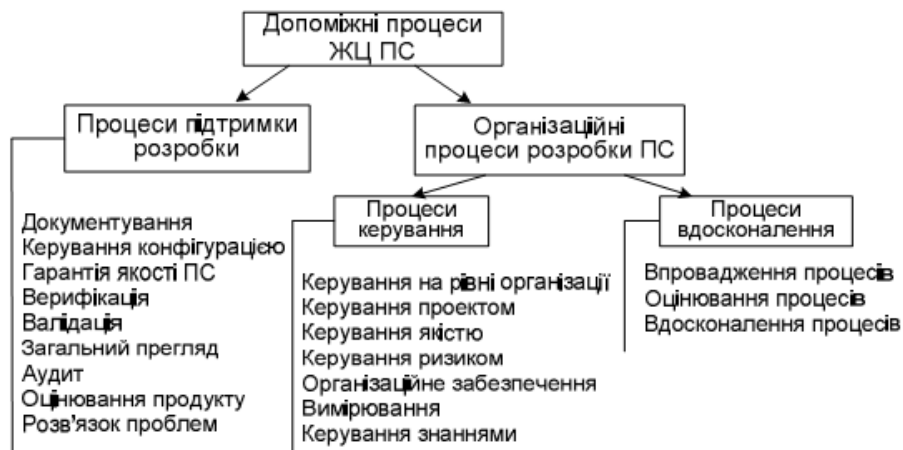


Рис. 2.2. Схема допоміжних процесів ЖЦ ПС

Ці процеси виконуються спеціальними службами, що здійснюють планування робіт у проекті, контроль процесів, визначення метрик для вимірювання продуктів, перевірку показників якості, дотримання стандартних положень та ін.

Процеси, дії і задачі наведені в стандарті в найбільш загальній природній послідовності. Залежно від цілей конкретного проекту головний розробник і менеджер вибирають процеси, дії і задачі, вибудовують визначену схему ЖЦ для застосування в цьому проекті.

Процеси керування в стандарті структуровані за рівнями і напрямками, вони жодним чином не зв'язані з існуючими методами і засобами програмної інженерії з розроблення ПС. Це дає можливість при їх виборі для застосування у ЖЦ зіставляти з ними звичні парадигми і методи розроблення (об'єктні, компонентні, сервісні й ін.) та засоби ядра знань SWEBOOK.

Таким чином, між стандартом ISO/IEC 12207 і ядром знань SWEBOOK існує зв'язок: вони взаємодоповнюють та збагачують один одного, більше у розробці відповідних документів брали участь одні ті самі висококваліфіковані фахівці в галузі програмування й інформатики. Інженерна дисципліна проектування ПС використовує теоретичні, прикладні методи і засоби розробки ПС і стандарти (ISO/IEC 12207, ISO/IEC 15404, ISO/IEC 9126 та ін.), а також рекомендації і методики керування розробкою, до яких відносять методи оцінки на процесах ЖЦ, якості ПС, витрачених ресурсів і вартості виконаних робіт. При цьому ядро знань SWEBOOK, а також численні монографії і статті рекомендують до застосування методи і засоби програмної інженерії, а стандарт дає настанови до побудови процесів на стандартизованій інженерній основі.

Природно, що в невеликих програмних проектах завжди можна буде застосовувати творчі і неформальні підходи, запропоновані фахівцями для створення різного роду унікальних продуктів, процес розроблення яких не завжди відповідає загальному стандарту.

Формування прикладних моделей життєвого циклу

Кожна ПС протягом свого існування проходить визначену послідовність *процесів* (процесів), починаючи від постановки задачі до її втілення в готову

програму, наступної експлуатації і остаточного виведення з експлуатації та списання. Така послідовність процесів називається *життєвим циклом* розробки ПС. На кожному процесі ЖЦ виконується визначена сукупність процесів і/або підпроцесів, кожний з яких породжує відповідний проміжний продукт, використовуючи при цьому результати попереднього процесу і доробок.

Модель життєвого циклу – це схема виконання робіт і задач у рамках процесів, що забезпечують розробку, експлуатацію і супровід програмного продукту. Ця схема відображає еволюцію ПС, починаючи від формулювання вимог і закінчуючи припиненням користування нею.

Історично така схема робіт містить у собі:

- розробку вимог або технічного завдання;
- розробку ескізного або технічного проекту;
- програмування або робоче проектування;
- пробну експлуатацію;
- супровід і поліпшення;
- зняття з експлуатації.

Основне призначення моделей ЖЦ є таким:

- планування і розподіл робіт і ресурсів між розробниками, а також керування програмним проектом;
- забезпечення взаємодії між розробниками проекту і замовником;
- спостереження і контроль робіт, оцінка проміжних продуктів ЖЦ на дотримання специфікацій вимог, правильне їх виконання, оцінка продукту і реальних витрат, у тому числі і щодо застосовуваних програмних засобів і інструментів;
- узгодження проміжних результатів із замовником;
- перевірка правильності кінцевого продукту шляхом його тестування на запланованих і погоджених із замовником наборах тестів;
- оцінка відповідності характеристик якості отриманого продукту заданим вимогам;
- обговорення використовуваних процесів ЖЦ з метою оцінки їх потенційних можливостей і недоліків, що виявлялися при їх застосуванні, а також визначення напрямів удосконалення або модернізації ЖЦ.

Отже, для побудови конкретної моделі ЖЦ ПС, що задовольняє концептуальній ідеї проектованої системи з урахуванням її складності і масштабу робіт, необхідно зробити правильний вибір процесів, їх завдань і дій відповідно до стандарту. На сьогодні основою формування нової моделі ЖЦ для конкретної прикладної системи є стандарт ISO/IEC 12207, що задає повний набір процесів (більш 40), охоплює всі можливі види робіт і завдань, пов'язаних з побудовою ПС, починаючи з аналізу предметної області і закінчуючи виготовленням відповідного продукту. Стандарт ISO/IEC 12207 надає загальний опис процесів на найвищому рівні, проте він не покликаний деталізувати виконання дій або задач, з яких складаються процеси. Він також не ставить вимоги до формату і змісту документів, що випускаються на різних процесах.

Процеси, дії і задачі наведені в стандарті в найбільш загальній природній послідовності, але це не означає, що в такій самій послідовності вони повинні бути застосовані в конкретній моделі ЖЦ ПС. Залежно від проекту процеси, дії і задачі стандарту вибираються, упорядковуються і включаються в модель ЖЦ. При

застосуванні вони можуть перекривати, переривати один одного, виконуватися ітераційне або рекурсивно. Це визначає «динамічний» характер стандарту і дозволяє реалізувати з його допомогою довільну модель ЖЦ ПС.

Тому організаціям, що будуть застосовувати стандарт у своїй роботі, знадобляться додаткові внутрішні стандарти або процедури, які визначатимуть різні деталі застосування вибраних елементів ЖЦ залежно від типу ПС. Крім цього, існують міжнародні стандарти з керування конфігурацією ПЗ, супроводу, документування, оцінювання якості, верифікації і валідації, тестування тощо.

Зі стандарту ISO/IEC 12207 можна вибирати тільки ті процеси, що найбільше підходять для реалізації конкретної ПС. Обов'язковими є основні процеси, що є у всіх відомих моделях ЖЦ. Залежно від цілей і задач предметної області вони можуть бути поповнені процесами з категорії організаційних процесів даного стандарту. Розробник приймає рішення щодо необхідності вміщення в нову модель ЖЦ засобів забезпечення якості компонентів, системи керування проектом або визначення набору процедур перевірки для забезпечення правильності продукту і відповідності його заданим вимогам.

Процеси, що включені в модель ЖЦ, призначені для реалізації стандартних задач процесів ЖЦ, вони можуть залучати інші процеси, що пов'язані із забезпеченням захисту даних. Інтерфейси (входи і виходи) будь-яких двох процесів ЖЦ повинні бути мінімальними і кожний з них повинен відповідати таким правилам:

- якщо процес А викликається процесом В і тільки процесом В, то А належить В;
- якщо функція викликається більше ніж одним процесом, то вона стає окремим процесом;
- перевірка будь-якої функції в ЖЦ є обов'язковою.

Іншими словами, якщо вирішення певної задачі потребує більше ніж один процес, то вона може сама набути статусу процесу, що використовується одно- або багаторазово протягом ЖЦ конкретної системи. Кожен процес повинен мати внутрішню структуру, встановлену відповідно до особливостей його виконання.

Процеси конкретної моделі ЖЦ орієнтовані безпосередньо на розробника даної системи. Розробник може виконувати один або кілька процесів і процес, у свою чергу, може бути виконаний одним або кількома розробниками. При цьому один з розробників має відповідати за один процес або за всі процеси моделі, навіть якщо окремі роботи виконує інший розробник.

Створювана модель ЖЦ узгоджується з конкретними методиками розробки систем і відповідними стандартами в області програмної інженерії, які існують або розробляються самостійно для проекту з урахуванням можливостей і особливостей ПС. Іншими словами, кожен процес ЖЦ підкріплюється вибраними для реалізації ПС засобами і методами програмування, а також методикою їх застосування і виконання.

При формуванні моделі ЖЦ важливу роль відіграють організаційні аспекти:

- структура колективу і зв'язків між ними;
- планування послідовності робіт і термінів їх виконання;
- підбір і підготовка ресурсів (людських, програмних і технічних) для виконання робіт;
- оцінка можливостей реалізації проекту в заданий термін, вартість і ресурси.

Впровадження моделі ЖЦ у практичну діяльність зі створення програмного продукту дозволить впорядкувати взаємини між суб'єктами процесу розроблення ПС і враховувати динамічні модифікації вимог до проекту і до системи.

Приклад. ЖЦ розробки ПС із задачами і діями процесу тестування.

Головне призначення процесу тестування ЖЦ – виконання задач процесу на основі входів (вхідні дані для виконання задач процесу) і виходів при завершенні задач, а також ролей і дій виконавців цих задач.

Відповідно до стандарту ISO/IEC 12207 задачі тестування розглянуті і розподілені за процесами ЖЦ ПС. Як результат, отримано єдиний безперервний процес тестування різних продуктів ПС, задачами якого є *підготовка, проведення й оцінювання* результатів тестування. Ці задачі розподілилися між 20 діями (кроками) процесу розроблення. Даний підхід до поглибленого тестування ПС доцільно застосовувати, наприклад, для систем реального часу.

На кроці *підготовки* здійснюється аналіз робочих продуктів процесу розроблення ПС (вхідних для даного кроку процесу тестування) для визначення цілей, об'єктів, сценаріїв і ресурсів тестування, адекватних кроку тестування. Результати виконання кроків підготовки тестування повинні фіксуватися в планах тестування (рис. 2.3).

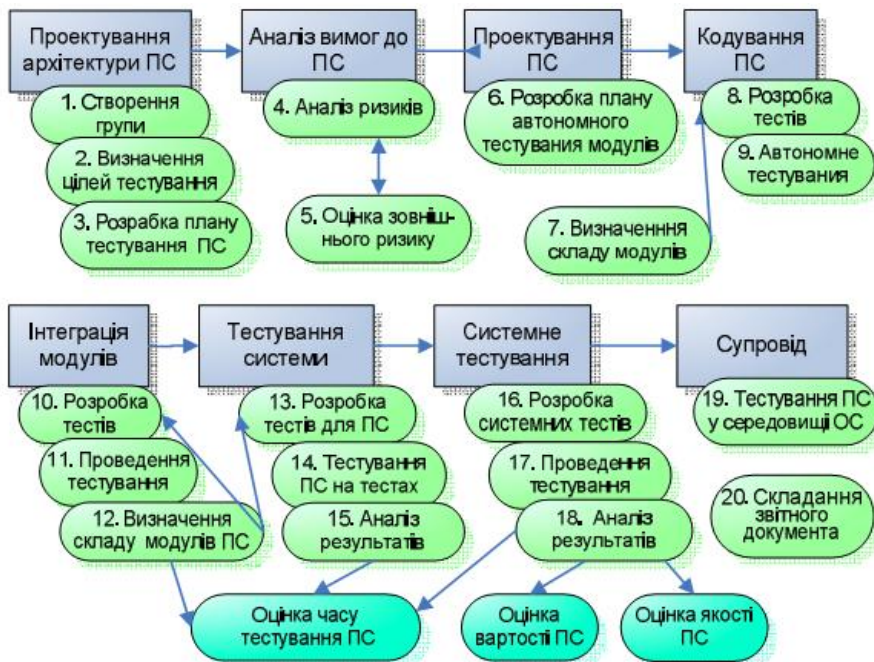


Рис. 2.3. ЖЦ з конкретними задачами на підпроцесах тестування ПС

На кроці *виконання* здійснюється фіксація результатів виконання тестів, їх порівняння з очікуваними результатами, визначення поточного стану робочого продукту ПС і прийняття рішення про достатність тестування.

Кожен крок процесу *розроблення* складається з набору розв'язуваних задач, їх розподілу за процесами і підпроцесами ЖЦ. Кроки процесу й окремих задач можуть виконуватися циклічно для різних об'єктів ПС при їх тестуванні.

Опис семантики задач і кроків процесу тестування наведено в табл.2.2.

Таблиця 2.2. Зміст задач процесу тестування

Крок процесу	Задачі процесу тестування
1. Створення групи тестування	1.1. Визначення учасників процесу тестування
	Розподіл обов'язків у групі і формування плану тестування
2. Аналіз ризику	2.1. Ідентифікація ризиків
	2.2. Упорядкування ризиків
	2.3. Розподіл ресурсів
3. Визначення цілей тестування	3.1. Ідентифікація цілей тестування
	3.2. Визначення критеріїв проходження тестів
	3.3. Упорядкування цілей тестування за оцінками ризику
4. Розроблення планів тестування	4.1. Розроблення плану тестування ПС
	4.2. Розроблення плану інтеграційного тестування
	4.3. Розроблення плану автономного тестування
	4.4. Розроблення плану комплексного тестування
5. Розроблення тестів	5.1. Проектування і розроблення тестів
	5.2. Підготовка тестових даних
	5.3. Перевірка тестових документів
6. Автономне й інтеграційне тестування	6.1. Автономне тестування модулів і аналіз результатів
	6.2. Інтеграційне тестування
	6.3. Повторне тестування після усунення дефектів
	6.4. Аналіз результатів інтеграційного тестування
7. Тестування ПС	7.1. Затвердження середовища і ресурсів тестування
	7.2. Тестування ПС
	7.3. Повторне тестування ПС після усунення дефектів
	7.4. Аналіз результатів завершення тестування ПС
	7.5. Тестування інсталяції ПС
8. Складання документа за тестуванням ПС і підготовка звіту	8.1. Збирання і аналіз даних про результати тестування
	8.2. Підготовка розв'язків і рекомендацій з використання ПС
	8.3. Підготовка кінцевого документа про результати тестування
	8.4. Перевірка рішень і підготовка документа звіту

Для підключення задач тестування до всіх процесів ЖЦ проводиться:

- розподіл обов'язків між учасниками процесу з урахуванням вимог щодо їх професійної підготовки;
- визначення стандартів на подання остаточних документів, метрик процесу, критеріїв початку і завершення задач і переходу до наступного кроку процесу;
- підбір методів тестування для вибраного класу ПС і перевірки правильності виконання задач тестування;
- розроблення спеціальних шаблонів для документування процесу тестування щодо кожного його кроку.

При завершенні тестування ПС з метою визначення часу тестування та вартості робіт враховуються результати і дані процесу розроблення, а також оформляється звітний документ на виготовлення ПС. Оцінювання ризику відмов проводиться на процесі підготовки тестування і на кроках аналізу, а оцінювання якості виконується після завершення тестування.

2. Типи моделей життєвого циклу ІТ-проектів

Розглянуті підходи щодо побудови різних видів моделей ЖЦ базуються на процесному підході до виконання програмних проектів. Вони використовувалися на практиці під час створення різних типів моделей ЖЦ, до яких належать такі моделі: каскадна, спіральна, інкрементна, еволюційна та ін.

2.1. Каскадна модель

Однією з перших почала застосовуватися *каскадна модель*, де кожна робота виконується один раз і в такому порядку, який подано на рис.2.4.

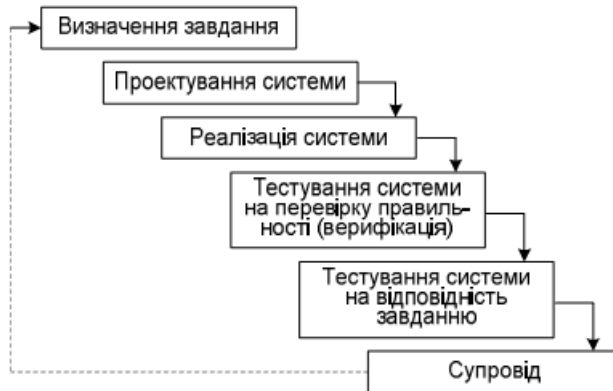


Рис. 2.4. Каскадна модель ЖЦ програмних систем

Тобто вважається, що кожна робота має бути виконана настільки ретельно, що після її закінчення і переходу до наступного етапу, повертатися до попереднього не буде потреби. Розробник перевіряє проміжний результат відомими методами верифікації і фіксує його як готовий еталон для наступного процесу.

Згідно з даною моделлю роботи і завдання процесу розроблення зазвичай виконуються послідовно, як це наведено у схемі.

Проте допоміжні і організаційні процеси (контроль вимог, керування якістю і ін.), як правило, виконуються разом з процесами розробки ПЗ. У даній моделі повернення до початкового процесу передбачається після супроводження і виправлення помилок.

Особливість такої моделі полягає у фіксації послідовних процесів розроблення програмного продукту. В її основу покладена модель фабрики, де продукт проходить стадії від задуму до виробництва, потім його передають замовнику у вигляді готового виробу, де заміна не передбачена, хоча можна подати інший подібний виріб. Недоліки цієї моделі такі:

- процес створення ПС не завжди вкладається в таку жорстку форму і послідовність дій;
- не враховуються змінювані потреби користувачів, нестабільні умови зовнішнього середовища, які впливають на зміни вимог до ПС під час і розроблення;
- значний розрив між часом внесення помилки (наприклад, на процесі проектування) і часом її виявлення (при супроводі), що призводить до суттєвої переробки ПС.

При застосуванні каскадної моделі можуть спостерігатися такі чинники ризику:

- вимоги до ПС недостатньо чітко сформульовані, або не враховують перспективи розвитку ОС, середовищ і т.п.;
- громізка система, що не допускає компонентної декомпозиції, може викликати проблеми щодо розміщення її в пам'яті або на платформах, не передбачених у вимогах;
- внесення швидких змін до технології і у вимоги може погіршити процес розроблення окремих частин системи або системи в цілому;
- обмеження на ресурси (людські, програмні, технічні і ін.) в ході розробки можуть звужити окремі можливості реалізації системи;
- отриманий продукт може виявитися не придатним для застосування внаслідок незрозуміння розробниками вимог або функцій системи або недостатньо проведеного тестування.

Переваги реалізації системи за допомогою каскадної моделі такі:

- всі завдання підсистем і системи реалізуються одночасно, завдяки чому не можна забути жодного завдання, а це сприяє встановленню стабільних зв'язків між ними;
- повністю розроблену систему з документацією на неї легше супроводжувати, тестувати, фіксувати помилки і вносити зміни не хаотично, а цілеспрямовано, починаючи з вимог, наприклад, додавати або замінювати деякі функції і повторювати процес.

Каскадну модель можна розглядати як модель ЖЦ, придатну для створення першої версії ПЗ з метою перевірки реалізованих в ній функцій. При супроводі і експлуатації можуть бути виявлені різного роду помилки, виправлення яких спричинить повторне виконання всіх процесів, починаючи з уточнення вимог.

2.2. Інкрементна модель

Цю модель (incremental) ще називають моделлю з нарощуванням або з приростом. Її суть полягає в розробці продукту ітераціями, і кожна ітерація закінчується випуском працездатної версії. У кожній новій версії додаються деякі функціональні можливості. Розробка системи починається з визначення набору всіх вимог до ПС і ділення процесу розроблення на ітерації. Кожна ітерація реалізується послідовно з використанням процесів ЖЦ і фіксації робочої версії системи, системи, що поступово наближається до остаточної версії (рис. 2.5).

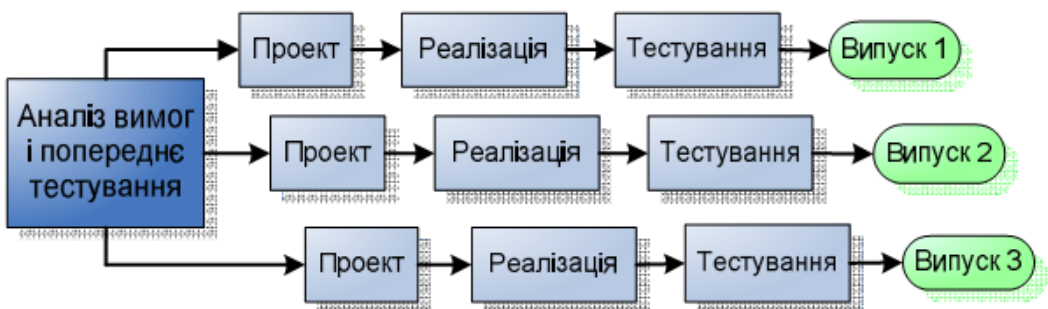


Рис. 2.5. Інкрементна модель ЖЦ

Перша проміжна версія системи, що створюється (випуск 1), реалізує частину вимог, у подальшу версію (випуск 2) додають додаткові вимоги до тих пір, поки не будуть остаточно виконані всі вимоги і розв'язані задачі розробки системи.

Для кожної проміжної версії на процесах ЖЦ виконуються необхідні процеси, роботи і завдання, зокрема, аналіз вимог і створення нової архітектури, які можуть бути виконані одночасно.

Процеси розроблення технічного проекту програмної системи, її програмування і тестування, збирання і кваліфікаційні випробування ПС виконуються при створенні кожної подальшої версії.

Відповідно до даної моделі ЖЦ, процеси якої практично такі самі, як і в каскадній моделі, наголос робиться на побудову деякої закінченої проміжної версії, а завдання процесу розроблення виконуються послідовно або частково паралельно для ряду окремих проміжних структур версії.

Роботи і завдання процесу розроблення наступної версії системи з додатковими вимогами або функціями можуть виконуватися неодноразово в одній тій же послідовності для всіх проміжних версій системи. Процеси супроводження і експлуатації можуть бути реалізовані разом з процесом розроблення версії шляхом перевірки частково реалізованих вимог в кожній проміжній версії і так до отримання кінцевого варіанта системи. Допоміжні і організаційні процеси ЖЦ зазвичай виконуються разом з процесом розроблення версії системи і до кінця розробки збираються дані, на підставі яких можна встановити рівень завершеності і якості виготовленої системи.

При застосуванні даної моделі необхідно враховувати такі чинники ризику:

– вимоги складені з урахуванням можливості їх зміни при реалізації продукту;

– всі можливості системи потрібно реалізувати одразу;

– швидка зміна технології і вимог до системи може призвести до порушення отриманої структури системи;

– обмеження в ресурсному забезпеченні (виконавці, фінанси) можуть призвести до несвоєчасного введення системи в експлуатацію.

Дану модель ЖЦ доцільно використовувати, у випадках, коли:

– бажано реалізувати деякі можливості системи швидко за рахунок створення проміжної версії продукту;

– система декомпонується на окремі складові частини, які можна реалізувати як деякі самостійні проміжні або готові продукти;

– можливе збільшення фінансування на розробку окремих частин системи.