

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**ЕКОНОМІКА ТА ОРГАНІЗАЦІЯ ВИРОБНИЦТВА ПРОГРАМНИХ
ПРОДУКТІВ**

**Конспект лекцій
для студентів технічних спеціальностей
усіх форм навчання**

Рекомендовано
редакційно-видавничою
радою університету,
протокол № 1 від 25.02.21 р.

Харків
НТУ «ХПІ»
2021

УДК 339.138

Р69

Рецензенти:

С. Е. Кучина, канд.екон.наук, доц.,

С. О. Васильцова, канд.екон.наук, доц., Національний технічний
університет «Харківський політехнічний інститут»

Романчик Т.В.

Р69 Економіка та організація виробництва програмних продуктів : конспект лекцій для студентів технічних спеціальностей усіх форм навчання / Романчик Т. В. – Харків : НТУ «ХПІ», 2021. – 76с.

У конспекті лекцій викладено теоретичні аспекти ефективної організації процесу виробництва програмного продукту. Програмний продукт розглядається як окрема категорія товарів, представлених на ринку. Розглянуто основні принципи та підходи до планування ходу виконання розробки, визначення вартісних показників матеріально-технічних ресурсів, що задіяні у процесі, розрахунку достатньої кількості працівників. Викладений матеріал дозволяє здійснити вартісну оцінку готового продукту та визначити економічну ефективність капіталовкладень у його розробку.

Конспект лекцій призначений для студентів ВНЗ спеціальності «Комп'ютерні науки», може бути корисним для фахівців у галузі розробки та просування складних програмних продуктів.

Табл. 6. Іл. 9 Бібліогр. 15 назв.

УДК 004.41

© Т.В. Романчик, 2021 р.

ЗМІСТ

Вступ	4
Тема 1. Поняття економіки програмного забезпечення	5
1.1 Основні поняття економіки програмного продукту	5
1.2 Організація економічно ефективного виробництва програмного продукту	7
1.3 Життєвий цикл програмного продукту	10
Тема 2. Основи ціноутворення на програмні продукти	20
2.1 Основні принципи ціноутворення на програмні продукти	20
2.2 Класифікація видів витрат на програмний продукт	32
2.3 Поняття собівартості програмного продукту	34
Тема 3. Методи оцінювання трудомісткості розробки програмного продукту	42
3.1 Трудомісткість обробки інформації: основні поняття	42
3.2 Види витрат праці при створенні програмного продукту	43
3.3 Методи оцінювання трудомісткості програмного продукту	45
Тема 4. Планування розробки програмних продуктів	48
4.1 Метод сітьового планування програмної діяльності	48
4.2 Застосування PERT-аналізу у плануванні проєктів	63
Тема 5. Економічна ефективність розробки програмного продукту	65
5.1 Критерії успішності проєкту	65
5.2 Прибуток та рентабельність розробки програмного продукту	66
5.3 Економічна ефективність капіталовкладень у програмний проєкт	68
5.4 Інвестиційна привабливість програмних проєктів	70
Список використаних джерел	73

ВСТУП

Дисципліна «Економіка та організація виробництва програмних продуктів» входить до переліку дисциплін, обов'язкових для вивчення студентами 4-го курсу спеціальностей «Прикладна математика» та «Комп'ютерні науки».

Предметом навчальної дисципліни «Економіка та організація виробництва програмних продуктів» є основні принципи та підходи до формування цінових характеристик програмних продуктів, чинники, що на них впливають, а також теоретичні підходи до раціональної організації виробництва програмних продуктів.

Дисципліна спрямована на отримання теоретичних знань та практичних навичок з питань економічної оцінки вартості розробки програмних продуктів та основних принципів організації їх виробництва.

Матеріал конспекту лекцій містить теоретичні аспекти ефективної організації процесу виробництва програмного продукту. Програмний продукт розглядається як окрема категорія товарів, представлених на ринку. Виробництво програмного продукту та його економічне оцінювання мають певні специфічні особливості. Розглянуто основні принципи та підходи до планування ходу виконання розробки, визначення вартісних показників матеріально-технічних ресурсів, що задіяні у процесі, розрахунку достатньої кількості працівників. Викладений матеріал дозволяє здійснити вартісну оцінку готового продукту та визначити економічну ефективність капіталовкладень у його розробку.

Конспект лекцій призначений для студентів ВНЗ спеціальності «Комп'ютерні науки», може бути корисним для фахівців у галузі розробки та просування складних програмних продуктів.

Тема 1. ПОНЯТТЯ ЕКОНОМІКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Основні поняття економіки програмного продукту

1.2 Організація економічно ефективного виробництва програмного продукту

1.3 Життєвий цикл програмного продукту

1.1 Основні поняття економіки програмного продукту

Професійне програмування (синонім виробництво програм) – діяльність, спрямована на отримання доходів за допомогою програмування.

Принциповою відмінністю від просто програмування є те, що є або, принаймні, передбачається деякий споживач, який готовий платити за використання програмного продукту. Звідси випливає важливий висновок про те, що професійне виробництво програм це завжди колективна діяльність, в якій беруть участь мінімум дві людини: програміст і споживач.

Програмний продукт – сукупність програм і супровідної документації по їх установці, налаштуванню, використанню і доопрацюванню.

При підготовці контракту на виробництво замовного програмного продукту замовнику і менеджерам необхідно оцінювати не тільки вимоги до функціональних характеристик програмного продукту, але і ресурси – економічні характеристики, які здатні задовольнити сторони. Для невеликих простих проєктів програмних продуктів у багатьох випадках достатньо достовірними можуть бути інтуїтивні оцінки необхідних економічних ресурсів, зроблені досвідченими керівниками, які реалізували кілька аналогічних проєктів. Однак інтуїтивні оцінки керівників щодо розмірів і складності великих програмних проєктів, як правило, відрізняються великими помилками при плануванні необхідних ресурсів і економічних характеристик.

Починаючи розробку великого проєкту, керівники, перш за все, повинні вирішити задачу – чи доцільно його створення і оцінити, якою буде можлива ефективність застосування готового продукту, чи виправдаються витрати на їх розробку і використання. Тому такі проєкти повинні починатися з прогнозування, аналізу та техніко-економічного обґрунтування (ТЕО) майбутнього життєвого циклу передбачуваного програмного продукту.

Інтуїтивні оцінки необхідних економічних ресурсів, які виконуються навіть досвідченими керівниками, що реалізували кілька аналогічних проєктів великих розмірів, складності, трудомісткості і термінів створення, як правило, відрізняються істотними недоліками і помилками внаслідок того, що людина в основному оптимістична, багатьом керівникам проєкт комплексу програм представляється меншим за розміром, більш простим і дешевим, що веде до первинних недооцінок його складності і до конфліктних ситуацій між замовниками і розробниками. Наслідком цього є великі помилки при плануванні економічних характеристик – термінів, трудомісткості і вартості створення програмних продуктів. Це призводить до значного запізнювання завершення розробок і перевищення ймовірних витрат ресурсів. Внаслідок знехтування ретельним економічним обґрунтуванням за деякими дослідженнями до 15 % проєктів складних програмних комплексів не доходить до завершення, а майже половина проєктів не вкладається у виділені ресурси, бюджет і терміни, не забезпечує необхідні характеристики якості. Типові ситуації, коли відставання термінів впровадження великих промислових систем управління та обробки інформації повністю залежить від неготовності для них програмних продуктів.

1.2 Організація економічно ефективного виробництва програмного продукту

При промисловому проектуванні і виробництві будь-якої продукції застосовуються такі основні економічні характеристики для оцінювання виробничих процесів і виробів:

– **вартість** – фінансові витрати на створення одного примірника або комплекту повноцінного продукту, який повністю задовольняє вимоги користувачів;

– **трудомісткість** – витрати праці колективу фахівців, що беруть участь в створенні готового продукту, який відповідає вимогам користувачів;

– **тривалість** – час, витрачений фахівцем або колективом на проектування і виробництво продукту, який задовольняє користувачів;

– **число фахівців** різної кваліфікації, які беруть участь у створенні продукту, що відповідає вимогам замовника.

Економічний аналіз виробництва програмного продукту в грошовому вираженні має ряд істотних особливостей, які обмежують його застосування при оцінці і прогнозуванні економічних показників таких проектів з наступних причин:

- підприємства, що створюють комплекси програм, мають значні відмінності в рівні заробітної плати фахівців, що не завжди адекватно відбивається на їх продуктивності праці і фінансових витратах;

- кожне підприємство має накладні витрати і податки, які можуть значно відрізнятись і ніяк не впливають на трудомісткість і тривалість безпосереднього виробництва програмного продукту;

- дуже різні оснащеності підприємств технологіями і засобами обчислювальної техніки, а також витрати на їх придбання та експлуатацію, які багаторазово використовуються для різних проектів;

- із загальних витрат на апаратуру і експлуатацію технологічних комп'ютерів і налагоджувальних стендів складно виділити частку, яку

необхідно включати у вартість виробництва конкретного програмного продукту.

Проте при укладенні контрактів на виробництво продуктів і для оцінки інтегральних економічних характеристик комплексів програм доводиться застосовувати величини витрат в грошовому вираженні, для чого розроблюються угоди (калькуляції) між замовником і розробником щодо подолання перерахованих труднощів при аналізі вартості виробництва програмних продуктів.

Сумарні витрати інтелектуальної праці фахівців на виробництво програмного продукту – **трудомісткість** – основним інтегральним економічним показником кожного програмного проєкту. Ці витрати підлягають оцінці і мінімізації за умови забезпечення заданих функціональних характеристик програмного продукту і його якості. На сукупну трудомісткість при створенні програмного продукту впливає ряд факторів, при визначенні яких на практиці використовуються різні одиниці. Трудомісткість характеризується часом продуктивної праці певного числа фахівців, необхідним для створення програмного продукту, його компонентів або виконання певного етапу робіт в життєвому циклі. Такий підхід призвів до активного використання одиниць трудомісткості: людино-день, людино-місяць, людино-рік (при цьому людино-рік передбачається складається в середньому з 250 робочих людино-днів з урахуванням вихідних і святкових днів). Подібні одиниці трудомісткості дозволяють зіставляти витрати в різних організаціях і навіть у різних країнах на аналогічні за розміром комплекси програм і не залежать від особливостей валюти, податків і подібних економічних чинників при оцінці вартості. Ці одиниці трудомісткості досить міцно увійшли в практику планування і оцінки процесів виробництва.

Тривалість виробництва програмного продукту залежить від багатьох факторів, і перш за все від його складності. Технологічний процес створення будь-яких комплексів програм включає ряд базових етапів, які

обов'язково доводиться реалізувати незалежно від витрат. Кожен етап вимагає деякого часу, що призводить для конкретних комплексів програм до відносно невеликих (в порівнянні з витратами праці) варіацій сумарної тривалості виробництва. Якщо розробка ведеться на досить високому технологічному рівні, то цикл виробництва складного програмного продукту принципово важко скорочувати без втрат його якості.

Кількість необхідних фахівців і особливості їх кваліфікації нерівномірно розподіляються по етапах життєвого циклу комплексу програм. На початкові етапи системного проектування і завершальні етапи випробувань програмного продукту потрібно відносно невелику кількість, але найбільш висококваліфікованих фахівців. На середніх етапах розробки і тестування програмних модулів і компонентів в проєкті бере участь найбільша кількість фахівців щодо невисокої кваліфікації. Однак для оцінок і прогнозування економічних характеристик широко застосовується **середнє число фахівців**. Йому відповідає середня продуктивність праці колективу при виробництві повністю нового складного комплексу програм, яка може служити орієнтиром для порівняння ефективності праці при створенні різних програмних продуктів. Ця характеристика різниться для різних класів, розмірів і інших параметрів комплексів програм, однак діапазон цих відмінностей не настільки великий, як зміни розміру, вимог до якості і інших параметрів.

Економічні показники для оцінювання виробництва програмних продуктів включають:

1. Статистичні дослідження економіки виробництва програмних продуктів:

– властивості і особливості основних економічних характеристик виробництва програмних продуктів;

– організацію і методику статистичних досліджень економічних характеристик програмних продуктів;

– основні складові витрат на виробництво програмного продукту;

- аналіз впливу факторів на економічні показники виробництва програмних продуктів;

- приклади економічних характеристик виробництва програмних продуктів;

2. Характеристики трудомісткості виробництва програмних продуктів:

- визначення цілей при оцінюванні трудомісткості виробництва комплексів програм;

- фактори, які доцільно враховувати при оцінці трудомісткості виробництва програмних продуктів;

- облік межі продуктивності, доступної розробнику-одиначці, рядків нових програм на рік;

3. Характеристики тривалості розробки програмних продуктів:

- особливості фіксування часу початку і закінчення виробництва програмного продукту;

- визначення меж можливого скорочення тривалості виробництва програмних продуктів;

- визначення меж нераціонально великих тривалостей виробництва комплексів програм;

4. Допоміжні характеристики виробництва програмних продуктів:

- середню продуктивність праці колективу при виробництві складного комплексу програм;

- оцінки необхідного середнього числа фахівців для виробництва конкретного програмного продукту.

1.3 Життєвий цикл програмного продукту

Життєвий цикл програми, програмної системи, програмного продукту включає в себе розробку, розгортання, підтримку і супровід. Якщо програмний продукт не коробковий, а досить складний, то його розгортання у клієнтів, як правило, реалізується окремими самостійними проектами

впровадження. Супровід включає в себе усунення критичних несправностей в системі і реалізується часто не як проєкт, а як процесна діяльність. Підтримка полягає в розробці нової функціональності, переробці вже існуючої функціональності, в зв'язку зі зміною вимог і поліпшенням продукту, а також усуненні некритичних зауважень до ПЗ, виявлених при його експлуатації (рис. 1.1). Життєвий цикл програмного продукту завершується виведенням продукту з експлуатації і зняттям його з підтримки і супроводу.

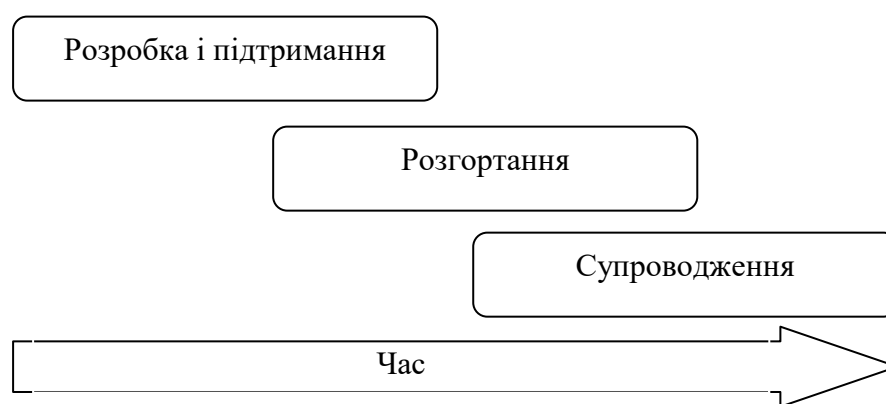


Рисунок 1.1 – Життєвий цикл програмного продукту

Програмний продукт визначається як набір комп'ютерних програм, процедур і, можливо, пов'язаної з ними документації і даних.

Процес визначається як сукупність взаємопов'язаних дій, що перетворюють деякі вхідні дані у вихідні. Кожен процес характеризується певними завданнями і методами їх вирішення, вихідними даними, отриманими від інших процесів, і результатами.

Кожен процес розділений на набір дій, кожна дія – на набір завдань. Кожен процес, дія або завдання ініціюється і виконується іншим процесом в міру необхідності, причому не існує заздалегідь визначених послідовностей виконання (природно, при збереженні зв'язків за вхідними даними).

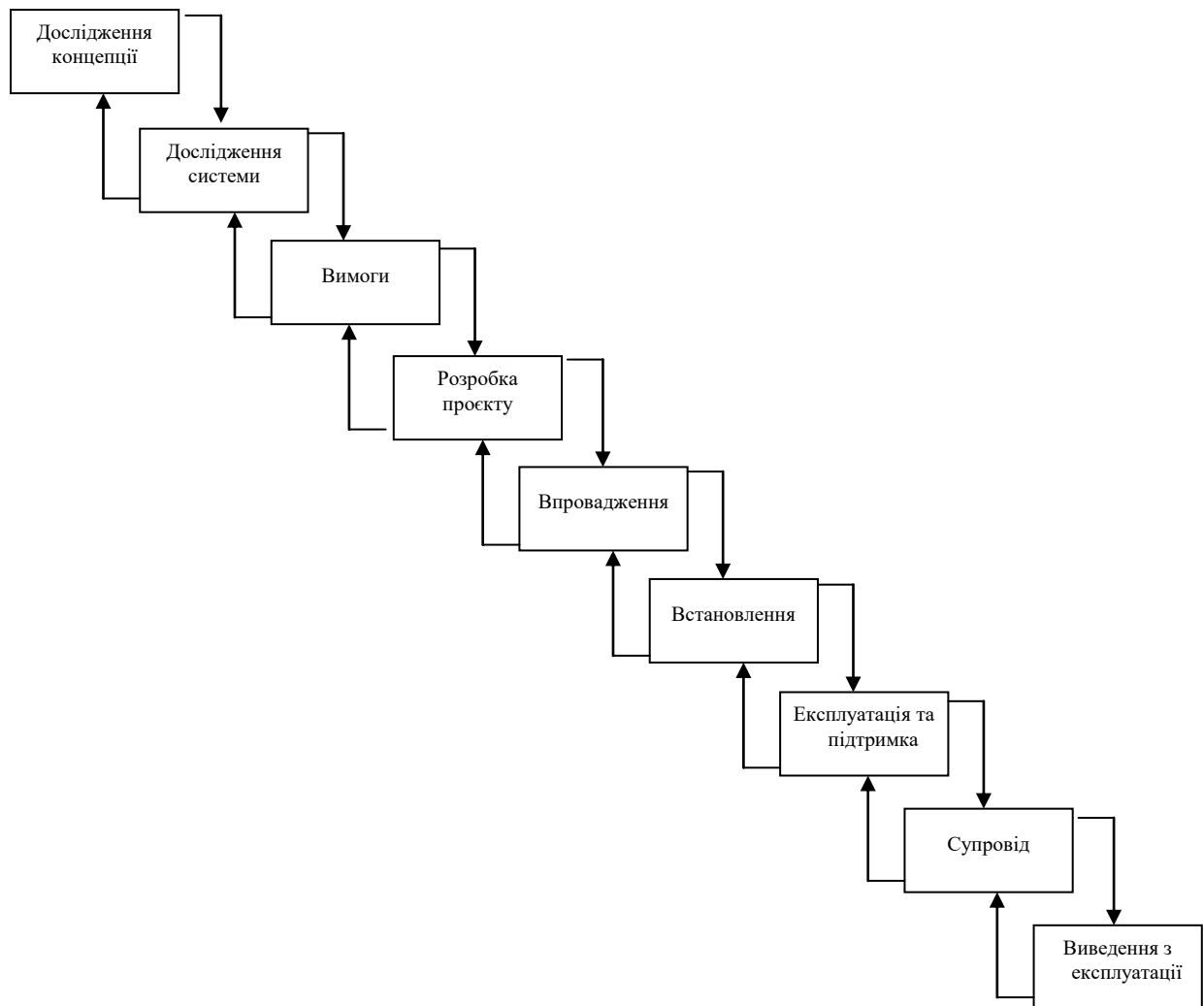


Рисунок 1.2 – Каскадна модель життєвого циклу розробки програмного забезпечення

Таблиця 1.1 – Стадії розробки, етапи та зміст робіт

Стадії розробки	Етапи робіт	Зміст робіт
1	2	3
I ТЕХНІЧНЕ ЗАВДАННЯ	Обґрунтування необхідності розробки програми	Постановка задачі. Збір вихідних матеріалів. Вибір і обґрунтування критеріїв ефективності та якості розроблюваної програми. Обґрунтування необхідності проведення науково-дослідних робіт.
	Науково-дослідні роботи	Визначення структури вхідних та вихідних даних. Попередній вибір методів вирішення завдань. Обґрунтування доцільності застосування раніше розроблених програм. Визначення вимог до технічних засобів. Обґрунтування принципової можливості вирішення поставленого завдання.

Закінчення табл. 1.1

1	2	3
	Розробка і затвердження технічного завдання	Визначення вимог до програми. Розробка техніко-економічного обґрунтування розробки програми. Визначення стадій, етапів і термінів розробки програми і документації до неї. Вибір мов програмування. Визначення необхідності проведення науково-дослідних робіт на подальших стадіях. Узгодження і затвердження технічного завдання.
II ЕСКІЗНИЙ ПРОЄКТ	Розробка ескізного проєкту	Попередня розробка структури вхідних та вихідних даних. Уточнення методів вирішення задачі. Розробка загального опису алгоритму розв'язання задачі. Розробка техніко-економічного обґрунтування.
	Затвердження ескізного проєкту	Розробка пояснювальної записки. Погодження та затвердження ескізного проєкту.
III ТЕХНІЧНИЙ ПРОЄКТ	Розробка технічного проєкту	Уточнення структури вхідних та вихідних даних. Розробка алгоритму розв'язання задачі. Визначення форми представлення вхідних і вихідних даних. Визначення семантики і синтаксису мови. Розробка структури програми. Остаточне визначення конфігурації технічних засобів.
	Затвердження технічного проєкту	Підготовка плану заходів щодо розробки та впровадження програм. Розробка пояснювальної записки. Узгодження і затвердження технічного проєкту.
IV РОБОЧИЙ ПРОЄКТ	Розробка програми	Програмування та налагодження програми.
	Розробка програмної документації	Розробка програмних документів відповідно до вимог ДСТУ.
	Випробування програми	Розробка, узгодження і затвердження програми та методики випробувань. Проведення попередніх державних (міжвідомчих) приймально-здавальних та інших видів випробувань. Коригування програми і програмної документації за результатами випробувань.
V ВПРОВАДЖЕННЯ	Підготовка і передача програми	Підготовка і передача програми і програмної документації для супроводу і (або) виготовлення. Оформлення та затвердження акту про передачу програми на супровід і (або) виготовлення. Передача програми в фонд алгоритмів і програм.

Примітка: допускається виключення 2-й стадії розробки, об'єднання 3-й і 4-й стадій, введення інших етапів робіт за погодженням із замовником. Необхідність проведення стадій розробки вказується в ТЗ.

Всі роботи, які можуть виконуватися в життєвому циклі програмних засобів, розподілені за п'ятьма основними, вісім'ю допоміжним і чотирма організаційними процесами.

Кожен процес життєвого циклу розділений на набір робіт; кожна робота розділена на набір завдань. Процес, робота або завдання по мірі необхідності ініціюються і виконуються іншим процесом, причому немає заздалегідь визначених послідовностей.

Основними процесами життєвого циклу є такі процеси:

- замовлення,
- поставка,
- розробка,
- експлуатація,
- супровід.

Процес замовлення складається з наступних робіт, виконуваних замовником:

- підготовка,
- підготовка заявки на підряд,
- підготовка і коригування договору,
- нагляд за постачальником,
- приймання та закриття договору.

Процес починається з визначення потреб замовника в програмному продукті або програмній послугі. Далі йдуть підготовка і випуск заявки на підряд, вибір постачальника і управління процесом замовлення аж до завершення приймання програмного продукту або програмної послуги.

Процес підготовки починається з опису потреби придбання готового ПП. Далі замовник формулює функціональні, комерційні, організаційні та споживчі властивості ПП, а також вимоги до безпеки, захисту та інші критичні вимоги поряд з вимогами до проєктування, тестування і відповідності стандартам і процедурам.

Процес поставки включає роботи, що виконуються постачальником: підготовку, підготовку відповіді, підготовку договору, планування, виконання та контроль, перевірку та оцінку, поставку та закриття договору.

Процес може бути розпочато з рішення про підготовку пропозиції у відповідь на заявку на підряд, надіслану замовником, або з підписання договору і вступу з замовником в договірні відносини з поставки програмного продукту або програмної послуги. Процес продовжується визначенням процедур і ресурсів, необхідних для управління і забезпечення проекту, включаючи розробку і виконання проектних планів з поставки ПП або програмної послуги замовнику.

Процес розробки включає роботи, що виконуються розробником: підготовку процесу; аналіз вимог до системи; проектування системної архітектури; аналіз вимог до програмних засобів; проектування програмної архітектури; технічне проектування програмних засобів; програмування і тестування програмних засобів; збірку програмних засобів; кваліфікаційні випробування програмних засобів; збірку системи; кваліфікаційні випробування системи; введення в дію програмних засобів; забезпечення приймання програмних засобів.

Процес включає роботи з аналізу вимог, проектування, програмування, збірки, тестування, введення в дію і приймання програмних продуктів. В даний процес можуть бути включені роботи, пов'язані з розробкою ПП, якщо це обумовлено в договорі. Розробник виконує або забезпечує виконання робіт по даному процесу відповідно до умов договору.

Процес експлуатації складається з наступних робіт, виконуваних групою впровадження замовника: підготовки процесу; експлуатаційних випробувань; експлуатації системи; підтримки користувачів.

Процес охоплює експлуатацію програмного продукту і підтримку користувачів в процесі експлуатації.

Процес супроводу складається з наступних робіт, виконуваних групою супроводу розробника (постачальника): підготовки процесу; аналізу проблем і змін; внесення змін; перевірки і приймання при супроводі; перенесення; зняття з експлуатації.

Даний процес реалізується при змінах (модифікаціях) програмного продукту і відповідної документації, викликаних проблемами, що виникли або потребами в його модернізації чи налаштуванні. Метою процесу є зміна існуючого програмного продукту при збереженні його цілісності. Даний процес охоплює питання переносу і зняття програмного продукту з експлуатації. Процес закінчується зняттям програмного продукту з експлуатації.

Допоміжні процеси життєвого циклу ПП сформульовані в ДСТУ наступним чином: процес документування; процес управління конфігурацією; процес забезпечення якості; процес верифікації; процес атестації; процес спільного аналізу; процес аудиту; процес вирішення проблем.

Процес документування є формалізованим описом документації, створюваної в процесі проектування ПП. Даний процес складається з набору робіт, за допомогою яких планують, проєктують, розробляють, випускають, редагують, поширюють і супроводжують документи, в яких мають потребу всі зацікавлені особи, такі як адміністратори, системні інженери і користувачі програмного продукту. Даний процес складається з наступних робіт: підготовки процесу; проєктування і розробки; випуску; супроводу.

Процес управління конфігурацією є процесом застосування адміністративних і технічних процедур на всьому протязі життєвого циклу програмних засобів (продуктів, комплексів, компонентів); позначення, визначення та встановлення стану програмних об'єктів; управління змінами і випуском об'єктів; опису та інформування про стан об'єктів і заявок на внесення змін; забезпечення повноти, сумісності та правильності об'єктів;

управління зберіганням, обігом і постачанням об'єктів. Даний процес складається з наступних робіт: підготовки процесу; визначення конфігурації; контролю конфігурації; обліку станів конфігурації; оцінки конфігурації; управління випуском і постачанням.

Процес забезпечення якості є процесом забезпечення гарантій щодо відповідності програмних продуктів і процесів у життєвому циклі проекту, встановленим вимогам і затвердженим планам. Забезпечення якості має бути організаційно і повноважно незалежним від суб'єктів, безпосередньо пов'язаних з розробкою програмного продукту або виконанням процесу в проєкті. Забезпечення якості може суб'єктивно (внутрішньо або зовні) залежати від того, демонструються докази якості продукту або процесу під управлінням постачальника або замовника. При забезпеченні якості можуть використовуватися результати інших допоміжних процесів, таких як верифікація, атестація, спільні аналізи, аудит і рішення проблем. Даний процес складається з таких робіт: підготовки процесу; забезпечення якості продукту відповідно до умов договору; забезпечення якості процесу (відповідності з умовами договору); забезпечення систем якості (відповідно ДСТУ).

Процес верифікації є процесом визначення ступеня відповідності функціонуючих програмних продуктів вимогам або умовам, сформульованим у договірних документах. Для оцінки ефективності витрат і виконуваних робіт верифікація повинна якомога раніше реалізовуватися у відповідних процесах, таких як постачання, розробка, експлуатація або супровід. Даний процес може включати аналіз, перевірку та випробування (тестування). Цей процес може виконуватися з різними ступенями незалежності виконавців. Ступінь незалежності виконавців може розподілятися як між різними суб'єктами в самій організації, так і суб'єктами в іншій організації з різними ступенями розподілу обов'язків. Даний процес називається процесом незалежної верифікації, якщо

організація-виконавець не залежить від постачальника, розробника, оператора або персоналу супроводу.

Процес атестації є процесом визначення повноти відповідності встановлених вимог створеного програмного продукту його функціональним призначенням. Атестація може проводитися на початкових етапах роботи. Даний процес може проводитися як частина роботи по забезпеченню приймання програмних продуктів. Умови реалізації даного процесу ідентичні умовам реалізації процесу верифікації.

Процес спільного аналізу є процесом оцінки станів і (при необхідності) результатів робіт за проектом. Спільні аналізи застосовуються як на рівні управління проектом, так і на рівні технічної реалізації проекту і проводяться протягом всього життєвого циклу договору. Даний процес може виконуватися двома будь-якими сторонами, які беруть участь в договорі, коли одна сторона (що аналізує) перевіряє іншу сторону (яку аналізують). Даний процес складається з наступних робіт: підготовки процесу; аналізу управління проектом; технічних аналізів. Процеси аналізу ходу робіт проводяться в терміни, встановлені проектним планом, або в строки, що визначаються зацікавленою стороною.

Процес аудиту є процесом визначення відповідності вимогам, планам та умовам договору. Він може виконуватися двома будь-якими сторонами, які беруть участь в договорі, коли одна сторона (що ревізує) перевіряє іншу сторону (яка ревізується). Аудиторські перевірки повинні проводитися в терміни, встановлені проектним планом (ами).

Процес вирішення проблем є процесом аналізу і вирішення проблем (включаючи виявлені невідповідності) незалежно від їх походження або джерела, які виявлені в ході виконання розробки, експлуатації, супроводу або інших процесів. Метою даного процесу є забезпечення способів своєчасного, відповідального і документованого аналізу, вирішення всіх виявлених проблем і визначення причин їх виникнення. При виявленні проблем (включаючи виявлені невідповідності) в програмному продукті або

роботі повинен бути підготовлений звіт з проблеми, що описує кожен виявлену проблему. Звіт з проблеми повинен бути складовою частиною вищеописаного процесу і охоплювати наступні питання: виявлення проблем і причин їх виникнення; дослідження, аналіз і вирішення проблем; визначення тенденцій, які обумовлюють виникнення проблем.

Організаційними процесами життєвого циклу є такі процеси: управління, створення інфраструктури, удосконалення та навчання.

Процес управління складається із загальних робіт, які можуть бути виконані як розробником, так і замовником, які керують відповідним(и) процесом(ами). Адміністратор відповідає за управління проектом, роботами і завданнями відповідного процесу або процесів, таких як замовлення, постачання, розробка, експлуатація, супровід або допоміжні процеси. Даний процес складається з таких робіт: підготовки та визначення області управління; планування; виконання і контролю; перевірки і оцінки; завершення.

Процес створення інфраструктури є процесом встановлення та забезпечення (супроводу) інфраструктури, необхідної для будь-якого іншого процесу. Інфраструктура може містити технічні та програмні засоби, інструментальні засоби, методики, стандарти і умови для розробки, експлуатації або супроводу. Даний процес складається з таких робіт: підготовки процесу; створення інфраструктури; супроводу інфраструктури.

Процес удосконалення є процесом ідентифікації, оцінки, вимірювання, контролю та поліпшення будь-якого процесу життєвого циклу програмного продукту. Даний процес складається з таких робіт: створення процесу; оцінки процесу; удосконалення процесу.

Процес навчання є процесом забезпечення початкового та продовженого навчання персоналу. Замовлення, постачання, розробка, експлуатація та супровід програмних продуктів в значній мірі залежать від кваліфікації персоналу. Наприклад, персонал розробника повинен бути відповідним чином навчений управлінню проектами та технології

програмування. Тому обов'язково має бути заплановано і заздалегідь виконано навчання персоналу з метою готовності його до робіт щодо замовлення, постачання, розробки, експлуатації або супроводу програмного проєкту. Даний процес складається з наступних робіт: підготовки процесу; розробки навчальних матеріалів; реалізації плану навчання.

Тема 2. ОСНОВИ ЦІНОУТВОРЕННЯ НА ПРОГРАМНІ ПРОДУКТИ

2.1 Основні принципи ціноутворення на програмні продукти

2.2 Класифікація видів витрат на програмний продукт

2.3 Поняття собівартості програмного продукту

2.1 Основні принципи ціноутворення на програмні продукти

При розробці програмного продукту під конкретне замовлення проблема зводиться до взаємної узгодженості думок між розробником і замовником за визначенням договірної ціни на розробку ПП.

У цьому випадку очевидно прагнення замовника знизити договірну ціну проєкту, від розробника же потрібно коректне обґрунтування її величини. Проблеми визначення договірної ціни полягають у тому, що процес створення ПП по суті є процесом перетворення специфікацій, вимог і побажань замовника у код і документацію програмного продукту. Складність управління таким процесом викликана складністю отримання кількісних оцінок вхідної і вихідної інформації, а також операцій по її перетворенню. Цей процес частково є творчим і важко піддається формалізації і моделюванню. В результаті точне планування ускладнено і тільки 35 % проєктів є успішними, 19 % – закінчуються невдало, 46 % проєктів реалізуються не в повному обсязі. Основні причини невдач пов'язані зі зміною бюджету проєкту – 45 %, порушенням календарних планів – 63 %, невідповідністю з узгодженими раніше функціональними можливостями. Таким чином, можна стверджувати, що наявність

ефективної методики оцінювання проєктів дозволить усунути значну частину причин зриву проєктів і сприятиме підвищенню конкурентоспроможності організації-розробника на ринку.

В основі існуючих методів розрахунку договірної ціни програмного продукту використовуються, як правило, такі показники:

- розміри програмної системи (в рядках коду),
- трудовитрати на розробку і впровадження,
- тривалість розробки,
- чисельність і кваліфікація фахівців розробника,
- нормативи оплати праці при створенні програмного коду,
- величина накладних витрат розробника, включаючи бажану норму прибутку.

Слід зазначити, що можливість отримання об'єктивних кількісних оцінок характеристик ПП корисна не тільки розробникам, але і замовникам. Обидві зацікавлені сторони можуть бачити, наскільки змінюється вартість проєкту при внесенні тієї чи іншої функціональності і визначити ступінь її важливості. За взаємною згодою з проєкту по розробці ПП може бути оперативно виключена функціональність, яка не виправдовує своєї ціни або реалізація якої може бути перенесена на наступну версію або навіть в інший проєкт.

Основна проблема розробника-виробника при оцінці ринкової вартості ПП зводиться до визначення ціни, що забезпечує максимальний дохід фірми. При цьому очевидно, що один і той же дохід можна отримати, продаючи товар за низькою ціною багатьом користувачам або за високою ціною обмеженій кількості користувачів, тобто ціна продажу та кількість продажів безпосередньо впливають один на одного. Перша стратегія ціноутворення розрахована на охоплення максимальної кількості можливих користувачів завдяки низькій ціні продукту, друга спрямована на реалізацію продукту найбільш платоспроможним користувачам по відповідно високій ціні. У чистому вигляді дані стратегії майже не

зустрічаються, найчастіше застосовуються їх модифікації. При другій стратегії після впровадження ПП на найбільш платоспроможному цільовому ринку користувачів ціни, як правило, знижуються з урахуванням платоспроможності наступної потенційної групи покупців. І навпаки, фірма довгий час може продавати свої пакети прикладних програм за низькою ціною, а в подальшому підвищити ціни, наприклад, за допомогою штучного розбиття продуктів на дві або більше складові частини.

Можна припустити, що при досить високій ціні на продукти доходи компаній можуть поступово зменшуватися. Здоровий глузд підказує, що, як правило, після продажу і установки ПП триває технічна підтримка користувача і з кожним новим користувачем трудовитрати розробника на підтримку збільшуються. Тому за інших рівних умов необхідно прагнути отримати максимальний дохід при меншій кількості користувачів. Разом з тим, чим більше користувачів користуються продуктом фірми, тим більше про нього говорять і опосередковано рекламують його, знижуючи тим самим витрати фірми на рекламу і просування. Таким чином, питання про те, що краще: – багато продажів по мінімальній ціні, або мало продажів за великою ціною – не має однозначної відповіді.

При постановці завдання ціноутворення, перш за все, необхідно визначити цілі цінової політики, тобто обрати один із напрямів розвитку компанії, який і складе основу стратегії ціноутворення.

1. Забезпечення виживаності ПП на ринках. Проблеми можуть виникнути через конкуренцію або зміни запитів споживачів. Щоб забезпечити розробку і продаж ПП, фірми змушені встановлювати низькі ціни в надії на доброзичливу відповідну реакцію споживачів, розуміючи, що в даний момент виживання важливіше прибутку. До тих пір, поки зниження ціни на ПП покриває витрати на його розробку, програмний продукт слід просувати, тобто продовжувати комерційну діяльність.

2. Максимізація прибутку. При реалізації даної мети фірми орієнтуються на рівень цін, що забезпечує в короткостроковому періоді

отримання максимального прибутку, і не розглядають довгострокові перспективи, зумовлені використанням інших елементів маркетингу, політикою конкурентів і регулюючою діяльністю держави.

3. Максимальне збільшення обсягу продажів. Збільшення обсягу збуту приведе до зниження витрат на одиницю продукції і збільшенню прибутку. Виходячи з можливостей ринку обирається «цінова політика наступу на ринок». Фірма знижує ціни на свою продукцію до мінімально допустимого рівня, підвищуючи частку свого ринку, добивається зниження витрат на одиницю продукції і на цій основі може і далі знижувати ціни. Але така політика приносить успіх, тільки якщо чутливість ринку до цін велика, якщо реально зменшуються витрати виробництва в результаті розширення обсягів виробництва і якщо ціни конкурентів не піддаються коригуванню в бік зменшення.

4. «Зняття вершків» завдяки встановленню високих цін. Фірма встановлює на новий продукт, що просувається на ринок, максимально можливу ціну завдяки порівняльним перевагам новинки. Коли продажі продукту за даною ціною скорочуються, фірма знижує ціну, залучаючи до себе наступний шар споживачів, досягаючи в кожному сегменті цільового ринку максимально можливого обороту.

5. Досягнення лідерства за якістю. Фірма, яка здатна закріпити за собою репутацію лідера за якістю, встановлює високу ціну, щоб покрити великі витрати, пов'язані із забезпеченням високої якості продукту.

Визначившись з цілями цінової політики, компанія повинна вибрати один або кілька методів визначення ціни на ПП. Класичне ціноутворення ґрунтується на визначенні фактичних витрат (собівартості) на виробництво і реалізацію продукту і додаванням до них бажаного прибутку. При виборі методів ціноутворення фірма виходить з того, що в будь-якому випадку мінімальна ціна визначається собівартістю продукту, а максимальна – наявністю якихось унікальних споживчих цінностей або монопольним становищем фірми на відповідному сегменті ринку.

Існуючі методи ціноутворення можна згрупувати наступним чином:

- методи ціноутворення, орієнтовані на витрати;
- методи ціноутворення, орієнтовані на попит;
- методи ціноутворення, орієнтовані на конкуренцію.

Методи ціноутворення, орієнтовані на витрати, засновані на обліку витрат виробника по розробці і реалізації продукту. В цьому випадку при визначенні ціни продажу необхідно виявити діапазон цін, що забезпечують покриття змінних і постійних витрат і отримання бажаного прибутку.

Одним з варіантів методу ціноутворення, орієнтованого на витрати, є **метод встановлення «цільової» ціни**, заснований на визначенні точки беззбитковості. В цьому випадку обрана цінова політика повинна забезпечити отримання «цільової» норми прибутку на зроблені витрати при встановлених обсягах продажів. Недолік цього методу пов'язаний з труднощами визначення величини обсягу продажів, так як сама ціна може істотно впливати на цей обсяг.

Методи ціноутворення, орієнтовані на попит, передбачають готовність споживачів оплачувати ПП за певною ціною (верхньою межею ціни). При цьому необхідно враховувати реакції споживачів на зміни цін і можливості їх диференціювання. При використанні даних методів не простежується безпосередній зв'язок між витратами і встановленням цін, за винятком випадків необхідності встановлювати ціни нижче рівня нижньої межі ціни.

Проблема в процесі використання даних методів полягає в тому, що попит значно важче визначити і висловити в кількісних величинах, ніж витрати. Оцінка попиту особливо ускладнюється відповідно до нових ПП, що просуваються на ринок, в силу відсутності статистики за минулі періоди.

Методи ціноутворення, орієнтовані на конкуренцію (конкурентів), ґрунтуються на порівнянні споживчих цінностей програмних продуктів розробника і конкурентів.

Точка зору користувача на споживчі властивості програмного продукту часто відрізняється від точки зору розробника. В цьому випадку при описі продукту розробнику необхідно дивитися на нього очима клієнта. Споживчі властивості продукту відіграють основну роль при порівнянні програмного продукту з конкурентами-аналогами, але при цьому не обов'язково, щоб продукт розробника перевершував ПП конкурентів за всіма параметрами. Необхідно вибрати головну конкурентну перевагу і на неї робити акцент.

Споживча цінність продукту залежить від ряду факторів, таких як якість продукту, можливість підключення інших додатків, платформонезалежна реалізація і т.ін.

Базова (вихідна) ціна, встановлена за допомогою одного з перерахованих методів ціноутворення, може коригуватися в бік зменшення наступними способами:

- формування договірної ціни на розробку;
- встановленням знижок при великому замовленні кількості ліцензій;
- пропозицією різних форм оплати (передоплатою, оплатою за фактом отримання продукту, оплатою з розстрочкою і т.ін.);
- встановленням знижок певним групам клієнтів;
- включенням в ціну продажу сервісів з технічного супроводу та підтримки користувача;
- наданням можливості вибору набору функціональностей програмного продукту і способу поставки (стандартної, професійної, промислової версії).

Формування договірної ціни на розробку програмного продукту

Техніко-економічне обґрунтування договірної ціни на розробку програмного продукту є процедурою оцінювання трудових, часових і фінансових ресурсів по створенню ПП, що відповідає вимогам замовника.

В основу визначення необхідних обсягів ресурсів повинні бути покладені:

- сукупність бізнес-процесів, що описують предметну область, і їх відносна важливість (пріоритет) для замовника;
- вимоги до функціональної повноти і якості реалізації кожного бізнес-процесу.

В якості основних показників оцінки вартості програмного продукту використовуються:

- складність (розміри);
- працевитрати на розробку;
- тривалість розробки системи в цілому і її окремих етапів;
- чисельність і кваліфікація фахівців, які залучаються до створення ПП;
- розміри фондів оплати праці фахівців на створення ПП в цілому і по кожному етапу життєвого циклу;
- інші прямі витрати і накладні витрати, пов'язані зі створенням ПП.

В основу визначення розмірів ПП покладено поняття «складності», що має на увазі кількість елементів програмної системи (програмних компонентів, файлів, вхідних і вихідних документів) і взаємозв'язків між ними.

В якості універсального вимірювача трудовитрат використовується показник «людино-місяць».

Кожен людино-місяць містить 160 людино-годин (чотири тижні, п'ять робочих днів, восьмигодинний робочий день).

Тривалість розробки і чисельність фахівців визначаються на основі працевитрат і нормативної продуктивності праці програміста, яка виражається в кількості рядків коду, що створюються програмістом в одиницю часу.

Прямий метод визначення розмірів програмного продукту на основі досвіду експертів

Створюваний програмний продукт слід декомпонувати до рівня елементарних компонентів, а для оцінки розмірів кожного компоненту використовувати або зовнішніх експертів, які мають досвід розробки подібних систем і готові прототипи, або залучати в якості експертів фахівців розробника і замовника.

Розміри програмного продукту визначаються у вигляді кількості рядків вихідного коду (Lines of code – LOC)..

Кожен з експертів повинен дати оптимістичну (o), песимістичну (p) і реалістичну (b) оцінки розмірності одного з елементів інтегрованого ПП. Середня оцінка по бета-розподілу визначається шляхом множення реалістичної оцінки на 4, додаванням оптимістичної і песимістичної оцінок і діленням отриманого результату на 6:

$$r_{ij}^k = (o_{ij} + 4b_{ij} + p_{ij}) / 6, \quad (2.1)$$

де r_{ij}^k – середня оцінка k -го експерту j -го програмного компоненту на i -му рівні.

Бланк експертного оцінювання розмірності програмного продукту

Язык програмування _____

Склад	Оцінки		
	Оптимістична	Реалістична	Пессимістична
1. Програмний продукт			
1.1. Програмний компонент 1			
1.2. Програмний компонент 2			
.....			

Після оцінювання всіх компонентів на кожному рівні, починаючи з нижнього, відбувається підсумовування результатів вимірювання за принципом «знизу-вгору»:

$$R = \sum_{k=1}^q \sum_{i=1}^n \sum_{j=1}^{m_i} r_{ij}^k / q, \quad k = \overline{1, q}, \quad i = \overline{1, n}, \quad j = \overline{1, m}, \quad (2.2)$$

де $k = \overline{1, q}$ – кількість експертів;

$i = \overline{1, n}$ – кількість рівнів декомпозиції програмного продукту;

$j = \overline{1, m}$ – кількість програмних компонентів на i -му рівні.

Визначення ринкової ціни програмного продукту

Для успішного ведення бізнесу необхідно не тільки прораховувати, скільки компанія заробить при досягненні запланованої кількості продажів, але і чітко уявляти, яку мінімальну кількість продажів необхідно для забезпечення беззбиткової роботи.

Завдання полягає у визначенні такої кількості продажів, нижче якого підприємство буде втрачати гроші, вище якого – заробляти.

Мінімально допустима кількість продажів, що покриває всі витрати на розробку першої версії програмного продукту, його просування на ринок і поставку споживачам, не приносячи при цьому ні прибутку, ні збитків, отримало назву «точка беззбитковості» (точка рівноваги). Таким чином, поняття точки беззбитковості є одночасно і певним критерієм ефективності діяльності фірми.

Компанія, котра визначила свою точку беззбитковості, може планувати відносно неї можливі варіанти обсягів продажів і очікувані результати отриманого прибутку. Точка беззбитковості є страховкою від отримання збитків. Якщо такий мінімальний обсяг продажів досягнутий, то підприємство покриває всі свої витрати, не отримуючи при цьому ніякого прибутку, але може прагнути до нарощування великих обсягів продажів. Якщо рівень обсягу продажів вище точки беззбитковості, то підприємство є

прибутковим, так як кожна одиниця продукції, яку продали понад, приносить прибуток. Якщо підприємство в продажу свого продукту не досягло точки беззбитковості, то таке підприємство збиткове і йому немає сенсу продовжувати свій бізнес в тому ж ритмі, так як при даній організації бізнесу без будь-яких змін його чекає банкрутство. Однак цей факт сам по собі не є однозначною причиною для припинення існування фірми. Для того щоб прийняти таке рішення, необхідно детально досліджувати структуру витрат фірми.

При визначенні рівня беззбитковості всі витрати поділяють на дві групи: VC (Variable Cost) – змінні витрати і FC (Fixed Cost) – постійні (фіксовані) витрати. До основних складових фіксованих витрат можна віднести, наприклад, такі види витрат:

- витрати на утримання апарату управління;
- оплату комунальних послуг, оренду приміщень;
- збільшення вартості основних засобів;
- амортизацію програмно-апаратних засобів (включаючи вартість першої версії ПП і придбаного системного ПО);
- виплати банківського кредиту.

Змінними витратами є витрати, що змінюються пропорційно кількості продажів, наприклад:

- основна зарплата фахівців відділів маркетингу і продажів (% від вартості тиражованого продукту);
- єдиний соціальний внесок;
- комплектуючі та витратні матеріали (картриджі, тонер, папір, диски CD-DVD і т. ін.) на тиражування продукту;
- накладні витрати відділів маркетингу і продажів (транспорт, послуги зв'язку, Інтернет, телефони і т. п.);
- витрати на відрядження.

Основним методом визначення точки беззбитковості є CVP-аналіз (Cast Value Profit – витрати, обсяг, прибуток), заснований на аналізі

співвідношень витрат, виручки і прибутку. У грошовому вираженні точка беззбитковості визначається за формулою

$$t_b = \frac{a}{(s \cdot x - bx) * s \cdot x}, \quad (2.3)$$

де x – кількість продажів;

s – ринкова ціна продажу одиниці продукції;

a – величина фіксованих витрат;

b – величина змінних витрат на одиницю продукції.

За рівнем точки беззбитковості можна судити про успішність компанії або погіршення її положення: чим нижче точка беззбитковості, тим легше підприємству отримати бажаний прибуток. Якщо значення точки беззбитковості збільшується, але масштаб підприємства не змінюється і обсяг реалізації продукції залишається стабільним, то це є тривожним сигналом для компанії. В даний момент слід виявити причини збільшення точки беззбитковості і вжити необхідних заходів для зупинки її зростання.

Чистий прибуток фірми визначається як різниця між виручкою і змінними і постійними витратами:

$$P = s \cdot x - (a + b \cdot x) = (s - b) \cdot x - a. \quad (2.4)$$

У цьому випадку кількість продажів, при якому досягається точка беззбитковості (прибуток фірми дорівнює нулю), визначається за формулою

$$x_0 = \frac{a}{(s - b)}. \quad (2.5)$$

Якщо обсяг ринку визначено (заданий), можна визначити ринкову ціну ПП при нульовому рівні прибутку:

$$s_0 = \frac{a + bx_0}{x_0}. \quad (2.6)$$

Якщо фірма хоче отримати додатковий прибуток (понад нормативної) і ринкова ціна відома, то кількість продажів при заданому рівні прибутку P_0 і ринковою ціною s_0 можна визначити за формулою

$$x_p = \frac{P_0 + a}{(s_0 - b)}. \quad (2.7)$$

Графічна інтерпретація визначення та аналізу точки беззбитковості представлена на рис. 2.1.

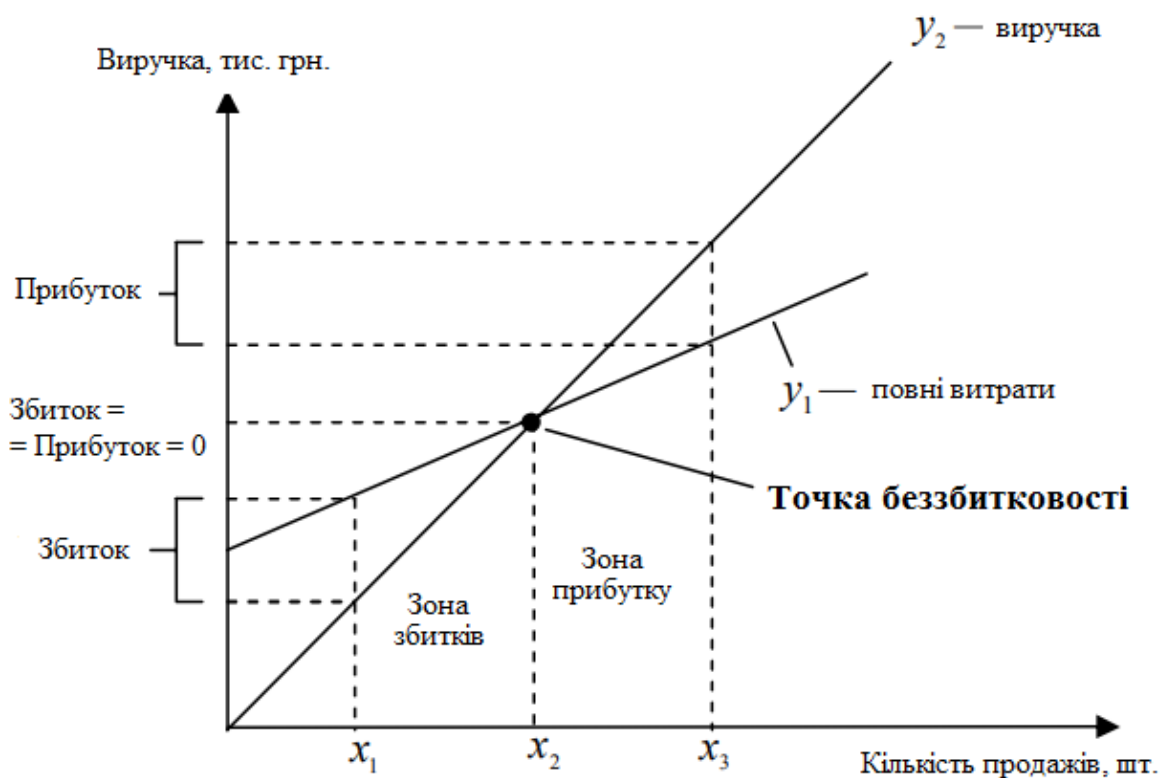


Рисунок 2.1 – Графічна інтерпретація визначення та аналізу точки беззбитковості

Фінансову стабільність компанії можна проаналізувати за допомогою показника **запасу міцності**:

$$Z = \frac{b \cdot x - t_b}{b \cdot x}. \quad (2.8)$$

Розраховуючи запас міцності, підприємство ставить собі за мету визначити межу, за якою починаються збитки. Запас міцності показує можливу величину зміни обсягу реалізації продукції (збільшення або зменшення), що дозволяє не перейти межу беззбитковості. Чим вище значення запасу міцності, тим стійкіше положення підприємства.

Коли точка беззбитковості досягнута, підприємство може розраховувати на отримання прибутку. Для визначення темпів зростання прибутку компанії використовується показник «виробничий важіль», що характеризує зростання прибутку компанії при збільшенні виручки на 1 %:

$$P_r = \frac{(s \cdot x - bx)}{(s \cdot x - b \cdot x - a)}. \quad (2.9)$$

Аналіз механізму прояву виробничого важелю дозволяє регулювати співвідношення змінних і постійних витрат з метою збільшення ефективності виробничої діяльності підприємства. Очевидно, чим вище постійні витрати компанії, тим складніше за інших рівних умов досягти точки беззбитковості.

2.2 Класифікація видів витрат на програмний продукт

Вартісна оцінка програмного забезпечення та визначення економічного ефекту у розробника припускає складання кошторису витрат, який включає наступні статті витрат:

- матеріали і комплектуючі (використовувані в процесі розробки і виробництва програмного забезпечення) (М);
- електроенергія (Е);
- основна заробітна плата розробників (З_о);

- додаткова заробітна плата розробників (Z_d);
- відрахування до фонду соціального страхування (C_{CH});
- амортизація обладнання і програмного забезпечення (A);
- спецобладнання для наукових (експериментальних) робіт (P_c);
- накладні витрати (P_n);
- інші прямі витрати (Π_3).

Перш за все, виділимо витрати, що залежать від обсягу виробленої продукції. Назвемо їх операційними (змінними) витратами. Критерій виділення змінних витрат: вони дорівнюють нулю, якщо даний продукт не виробляється зовсім (Витрати праці програмістів залежать від обсягу і складності програми, що розробляється. І якщо ми розробляємо дві схожі програми, то праці на них витратимо, швидше за все, в два рази більше, ніж на одну програму).

Всі інші витрати залежать від структури підприємства, його обладнання, постійного персоналу, зобов'язань підприємства і т.ін. Назвемо їх постійними витратами.

Наступний за зручністю виділення вид витрат – витрати, пов'язані з виробництвом тільки даного продукту. Наприклад, придбання ліцензійного ПЗ, використовуваного тільки нашою програмою. Очевидно, що всю вартість цього ПЗ потрібно віднести до нашої програми. Особливістю двох перших видів витрат є те, що їх безпосередньо можна віднести на виробництво даної продукції (проєкту). Їх сума називається прямі витрати.

Далеко не всі витрати можна повністю віднести на один проєкт. Наприклад, базове програмне забезпечення буде використовуватися не тільки в нашому проєкті, а й в інших, тому його вартість не можна повністю віднести на один проєкт. Назвемо такі витрати загальними витратами. Для визначення частки загальних витрат, що припадають на один проєкт доводиться робити припущення і домовлятися про принципи розподілу.

Нарешті, само утримання фірми вимагає витрат, не пов'язаних безпосередньо з виробництвом. До них відносяться витрати на утримання

приміщень, зарплату АУП і обслуговуючого персоналу, виплата податків, штрафів і т.п. Такі витрати називаються накладними витратами. Звичайною практикою віднесення частини накладних витрат на конкретний проєкт є визначення відсотка. Бухгалтерія фірми розраховує обсяг загальнофірмових витрат, ділить його на загальний обсяг виробництва і повідомляє виконавцям, яку частину (відсоток) зароблених ними грошей доведеться витратити на загальнофірмові потреби.

2.3 Поняття собівартості програмного продукту

Наведена вище класифікація витрат дозволяє застосувати відносно прості методики для їх оцінювання. Розглянемо особливості розрахунків кожного виду витрат.

Операційні витрати

Це найбільш простий для розрахунку вид витрат. У більшості практично важливих випадків витрати обчислюються як добуток вартості одиниці ресурсу на кількість спожитого ресурсу. У таблиці 1 зведені основні види операційних витрат, що зустрічаються при розробці програмного забезпечення.

Таблиця 2.1 – Основні види операційних витрат

Види операційних витрат	Методика розрахунку
Зарплата при погодинній оплаті праці	Витрати = (фактичний час) * (погодинна ставка) * (1 +% відрахувань)
Матеріали	Витрати = (кількість використаних матеріалів) * (вартість одиниці)
Послуги	Витрати = (обсяг послуг) * (вартість одиниці послуги)

Стаття «Основна заробітна плата»

Основною статтею витрат на створення програмного забезпечення є заробітна плата розробників (виконавців) проєкту, в число яких прийнято включати інженерів-програмістів, що беруть участь в написанні коду,

керівників проєкту, системних адміністраторів, дизайнерів, які розробляють призначений для користувача інтерфейс, розробників баз даних, Web-майстрів та інших фахівців, необхідних для вирішення спеціальних завдань в команді.

Оплата праці здійснюється на основі Єдиної тарифної сітки (ЕТС), в якій представлені тарифні розряди і тарифні коефіцієнти з урахуванням категорії, посади, освіти, складності виконуваної роботи.

За даними про специфіку та складність виконуваних функцій складається штатний розклад групи фахівців-виконавців, що беруть участь в розробці програмного забезпечення, з визначенням освіти, спеціальності, кваліфікації та посади.

Основна заробітна плата (Z_o) є основною статтею витрат при розробці програмного забезпечення і розраховується за формулою

$$Z_o = TC_i K_i n t, \quad (2.10)$$

де TC_i – тарифна ставка фахівця i -го розряду, грн.

K_i – коефіцієнт преміювання (встановлюється на підприємстві);

n – кількість виконавців, що брали участь в розробці програмного забезпечення, чол;

t – кількість часу, витраченого на розробку програмного забезпечення, місяців.

Тарифна ставка i -го розряду (TC_i) визначається шляхом множення діючої тарифної ставки 1-го розряду (TC_1) на тарифний коефіцієнт (ТК), що відповідає встановленому тарифному розряду відповідно до Єдиного тарифно-кваліфікаційного довіднику.

Тарифна ставка (TC_i) визначається за формулою

$$TC_i = TC_1 TK, \quad (2.11)$$

де TC_1 – діюча на підприємстві тарифна ставка першого розряду, грн;

ТК – тарифний коефіцієнт згідно ЄТКД.

Стаття «Додаткова заробітна плата» (З_д) на конкретне програмне забезпечення включає виплати, передбачені законодавством про працю (оплата відпусток, пільгових годин і інших виплат, не пов'язаних з основною діяльністю виконавців), і визначається за нормативом у відсотках до основної заробітної плати:

$$З_{д} = \frac{З_{о} \cdot Н_{д}}{100\%}, \quad (2.12)$$

де $З_{д}$ – додаткова заробітна плата виконавців на конкретне програмне забезпечення, грн.

$Н_{д}$ – норматив додаткової заробітної плати на підприємстві, %.

Стаття «Єдиний соціальний внесок»

Єдиний соціальний внесок визначається згідно з чинним законодавством на відповідний рік, за нормативом, у відсотковому відношенні до фонду основної та додаткової заробітної плати виконавців.

До даних витрат відносять:

- відрахування до Фонду соціального страхування від нещасних випадків;
- відрахування до Фонду соціального страхування на випадок тимчасової втрати працездатності;
- відрахування в пенсійний фонд
- відрахування до фонду соціального страхування на випадок безробіття.

Ставки єдиного соціального внеску встановлюються відповідно до класів професійного ризику виробництва, до яких віднесено платників єдиного внеску, з урахуванням видів їх економічної діяльності.

Для розробників програмного продукту ставка єдиного соціального внеску приймається в розміре 22 % від суми основної ($З_{о}$) і додаткової ($З_{д}$) заробітної плати.

Сума відрахувань на соціальні потреби розраховується за формулою

$$Z_{\text{сз}} = \frac{(Z_{\text{д}} + Z_{\text{о}}) N_{\text{ЕВВ}}}{100\%}, \quad (2.13)$$

де $N_{\text{ЕВВ}}$ – норматив відрахувань в Єдиний соціальний внесок, в%.

Стаття «Матеріали і комплектуючі» (М) – відображає витрати на матеріали та приладдя, необхідні для проведення науково-дослідних робіт, магнітні носії, папір, тонер та інші матеріали, необхідні для розробки програмного забезпечення. Витрати визначаються за діючими відпускними цінами.

Стаття «Електроенергія» (Е)

Витрати на електроенергію знаходяться, виходячи з тривалості періоду розробки програмного забезпечення, кількості кВт/г, витрачених на його проектування і тарифу за 1 кВт / год. за такою формулою

$$\mathcal{E} = K_{\text{е}} T_{\text{р}} \quad (2.14)$$

де $K_{\text{е}}$ – вартість одного кВт / год, грн;

$T_{\text{р}}$ – кількість кВт / год.

Тарифи на електроенергію застосовуються відповідно до тарифів на електроенергію на відповідний період часу, коли розробляється програмне забезпечення.

Постійні витрати

Витрати на обладнання

Стаття «Амортизація основних засобів та нематеріальних активів»

Організація і функціонування будь-якого виробництва вимагає певних ресурсів. Виділяють матеріальні (будівлі, споруди, обладнання, придбане для виконання проєкту) і нематеріальні ресурси (об'єкти інтелектуальної власності).

Як правило, обладнання, придбане для виконання проєкту, буде використовуватися і після його завершення. Тому повністю відносити його вартість на один проєкт не можна.

Для визначення частки вартості обладнання, віднесеної до конкретного проєкту, використовується метод нарахування амортизації.

Розробка програмного забезпечення пов'язана з використанням ліцензійних операційних систем, програмних середовищ і утиліт, графічних зображень. Всі вони є об'єктами інтелектуальної власності та їх неліцензійне («піратське») використання незаконно. Придбання прав на використання таких об'єктів не супроводжується передачею покупцеві будь-яких матеріальних об'єктів. Тому, в бухгалтерському обліку права на використання об'єктів інтелектуальної власності реєструються як «Нематеріальні активи» [3].

Якщо нематеріальні активи входять в специфічні структурні витрати (тобто не використовуються ні де, крім нашого проєкту), вся їх вартість відноситься на вартість проєкту. Якщо вони використовуються декількома проєктами, або планується їх використання після закінчення проєкту, застосовується метод нарахування амортизації.

Права на використання об'єктів інтелектуальної власності можуть бути безстроковими або обмеженими в часі [4]. Період амортизації, природно, не може перевищувати строк прав (використання об'єкта після закінчення терміну незаконно). Однак період амортизації може бути менше цього терміну (наприклад, у зв'язку з моральним старінням використовуваного об'єкта інтелектуальної власності).

Витрати за статтею «Амортизація основних засобів та нематеріальних активів» розраховуються по одному з методів нарахування амортизації (лінійний, нелінійний, продуктивний), виходячи з доцільності його застосування.

Амортизація нараховується на всі основні засоби і нематеріальні активи, що знаходяться на балансі підприємства, незалежно від характеру їх участі у виробничому процесі.

Грошовий вираз амортизації є амортизаційним відрахуванням, яке входить в поточні витрати. Величина амортизаційних відрахувань визначається на основі норм амортизації.

Норма амортизації – це встановлений розмір амортизаційних відрахувань на повне відновлення, виражене у відсотках. Норма амортизації встановлюється на основі економічно доцільного терміну служби і повинна забезпечити відшкодування зносу основних засобів до моменту можливого їх морального і фізичного зносу і створити економічну основу для їх заміни.

Норма амортизації для лінійного способу нарахування обчислюється за формулою

$$H_a = \frac{100\%}{T_c}, \quad (2.15)$$

де T_c – термін служби устаткування, років.

За статтею «Амортизація основних засобів та нематеріальних активів» розраховуються амортизаційні відрахування (A), виходячи з вартості основних засобів (C), що використовуються в процесі розробки програмного забезпечення, термінів експлуатації обладнання (T_c) і річної норми амортизації (H_a).

Для лінійного способу нарахування амортизаційні відрахування рівномірно розподілені на весь період служби обладнання і обчислюються на один рік.

$$A = \frac{C \cdot H_a \cdot T_c}{12}. \quad (2.16)$$

Спрощений спосіб нарахування витрат на обладнання:

Визначається ресурс обладнання (наприклад, в роках роботи). Загальна вартість обладнання ділиться на ресурс. Результат – вартість одиниці ресурсу. Визначається обсяг ресурсу обладнання, необхідний для реалізації проекту. Частка вартості ресурсу, віднесена на проєкт дорівнює вартості одиниці ресурсу помноженої на необхідний обсяг.

Засоби обчислювальної техніки застарівають досить швидко. Основна причина цього – технічний прогрес. Нові покоління техніки настільки перевершують попередні, що використовувати техніку 2-3 річної давнини вже не має ніякого сенсу. Тому для обчислювальної техніки призначаються скорочені терміни амортизації (2-3 роки.)

Стаття «Витрати на спецобладнання» (P_c) включає витрати на придбання допоміжних технічних і програмних засобів спеціального призначення, необхідних для розробки конкретного програмного забезпечення, включаючи витрати на їх проєктування, виготовлення, налагодження, встановлення та експлуатацію. Витрати розраховуються за формулою

$$P_c = \sum_{i=1}^n Ц_c, \quad (2.17)$$

де $Ц_c$ – вартість конкретного спеціального обладнання, грн;

n – кількість застосовуваного спеціального обладнання.

Стаття «Накладні витрати» (P_n), включає витрати, пов'язані з необхідністю утримання апарату управління, допоміжних господарств і дослідних (експериментальних) виробництв, а також з витратами на загальногосподарські потреби (P_n).

Накладні витрати, які відносяться на конкретне програмне забезпечення, розраховуються у відсотковому відношенні до основної заробітної плати виконавців, за формулою

$$P_n = Z_o \frac{H_p}{100\%}, \quad (2.18)$$

де P_n – накладні витрати на конкретне програмне забезпечення (грн.);

N_p – норматив накладних витрат в цілому по організації застосовується в розмірі 10 % – 100 %.

Стаття «Інші прямі витрати» (ПЗ) на конкретне програмне забезпечення включає витрати на придбання та підготовку спеціальної науково-технічної інформації та спеціальної літератури.

На підставі отриманих даних за окремими статтями витрат розраховується загальна сума витрат на розробку програмного забезпечення.

Загальна сума витрат за кошторисом (планова собівартість) (С) на програмне забезпечення розраховується за формулою

$$C = M + \text{Э} + Z_o + Z_d + Z_{cz} + A + P_c + P_n + P_z . \quad (2.19)$$

Розрахунок ціни програмного продукту

Замовник оплачує розробнику всю суму витрат за проектом, включаючи прибуток. Після сплати податків з прибутку в розпорядженні розробника залишається чистий прибуток від проекту. З огляду на те, що програмне забезпечення розробляється для одного об'єкта, чистий прибуток можна вважати економічним ефектом розробника від реалізованого програмного забезпечення.

Відпускна ціна продукції формується, виходячи з планової собівартості виробництва продукції, всіх видів встановлених податків і прибутку, а також якості, споживчих властивостей продукції і кон'юнктури ринку.

Відпускна ціна на продукцію розраховується за формулою

$$OЦ = C + П , \quad (2.20)$$

де $OЦ$ – відпускна ціна розробника, грн;

C – планова собівартість, грн;

$П$ – прибуток, грн.

Прибуток розраховується за такою формулою

$$\Pi = \frac{C \cdot R}{100\%} \quad (2.21)$$

де R – рівень рентабельності, %.

Вартість проекту з урахуванням ПДВ являє собою суму відпускнуї ціни та податку на додану вартість.

$$\text{ПДВ} = \frac{(C + \Pi) \cdot \text{Ставка ПДВ \%}}{100\%} . \quad (2.22)$$

Тема 3. МЕТОДИ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

3.1 Трудомісткість обробки інформації: основні поняття

3.2 Види витрат праці при створенні програмного продукту

3.3 Методи оцінювання трудомісткості програмного продукту

3.1 Трудомісткість обробки інформації: основні поняття

У реальній проєктній ситуації досить важко виділити і підрахувати величину трудомісткості як для реально діючої, так і для проєктованої системи.

Оцінка трудомісткості є одним з найбільш важливих видів діяльності в процесі створення програмного проєкту.

Недооцінка вартості, часу і ресурсів, необхідних для створення програмного проєкту, тягне за собою недостатню чисельність проєктної команди, надмірно стислі терміни розробки і, як результат, втрату довіри до розробників у разі порушення графіку. З іншого боку, перестраховка і переоцінка можуть виявитися нітрохи не краще. Якщо для проєкту виділено більше ресурсів, ніж реально необхідно, причому без належного контролю за їх використанням, то ні про яку економію ресурсів годі й казати. Такий

проект виявиться більш дорогим, ніж повинен був бути при грамотній оцінці, і призведе до запізнювання з початком наступного проекту.

3.2 Види витрат праці при створенні програмного продукту

Витрати праці залежать від складності та трудомісткості кожного з етапів життєвого циклу створення програмного продукту:

- аналізу предметної області та розробки вимог до програмної системи;
- проектування;
- програмування;
- тестування і комплексних випробувань;
- дослідної експлуатації.

У реалізації проекту на кожному етапі беруть участь три групи фахівців:

- 1) керівник проекту, системні аналітики;
- 2) безпосередні розробники програмних систем і фахівці з комплексування;
- 3) технічний персонал, що забезпечує тестування, документування та дослідну експлуатацію ПП.

Середня чисельність фахівців Z , які повинні бути залучені до реалізації проекту, визначається за формулою

$$Z = T / Д. \quad (3.1)$$

де T – рацевитрати;

$Д$ – тривалість розробки.

В основу визначення фонду оплати праці покладені:

- 1) тривалість реалізації кожного етапу життєвого циклу розробки програмного продукту;
- 2) кількість і якісний склад фахівців, що залучаються до кожного етапу розробки;

3) базова місячна ставка фахівця-програміста.

У табл. 3.1 і 3.2 наведені середньостатистичні розподіли перших двох величин за основними етапами життєвого циклу створення програмного продукту.

Використовуючи ці розподіли, за формулою (3.2) можна розрахувати середню чисельність співробітників, зайнятих на кожному з етапів створення програмного продукту:

$$Z_i = \alpha_i T / \beta_i D, \quad i = \overline{1,4}. \quad (3.2)$$

У табл. 3.1 дано відносний розподіл чисельності фахівців на кожному з чотирьох етапів життєвого циклу створення програмного продукту.

Таблиця 3.1 – Розподіл працевитрат та тривалість розробки проєкту за основними етапами життєвого циклу створення програмних систем

Етапи життєвого циклу	Працевитрати α , %	Тривалість розробки β , %
1. Аналіз предметної області та розробка вимог	8–10	10–20
2. Проєктування	14–22	20–30
3. Програмування	41–46	32–35
4. Тестування та комплексне випробування	24–27	20–25

Таблиця 3.2 – Розподіл фахівців за етапами життєвого циклу програмного продукту

Етапи життєвого циклу	Типи фахівців		
	Аналітики	Програмісти	Технічні фахівці
1. Аналіз предметної області та розробка вимог	40	20	40
2. Проєктування	35	35	30
3. Програмування	10	65	25
4. Тестування та комплексне випробування	15	60	25

Чисельність кожного типу фахівців на кожному з етапів життєвого циклу створення програмної системи визначається за наступним виразом:

$$Z_{ij} = P_{ij} \cdot Z_i, i = \overline{1,4}; j = \overline{1,3}, \quad (3.3)$$

де P_{ij} – відносна частка фахівців j -го типу, що залучаються для реалізації проєкту на i -му етапі, %.

3.3 Методи оцінювання трудомісткості програмного продукту

Існують різні методи оцінки програмних проєктів.

1. Алгоритмічне моделювання. Метод заснований на аналізі статистичних даних про раніше виконані проєкти, при цьому визначається залежність трудомісткості проєкту від якогось кількісного показника програмного продукту (ПП) (зазвичай це розмір програмного коду). Проводиться оцінка цього показника для даного проєкту, після чого за допомогою моделі прогнозуються майбутні витрати.

2. Експертні оцінки. Проводиться опитування декількох експертів за технологією розробки ПП, які знають область його застосування. Кожен з них дає свою оцінку трудомісткості проєкту, після чого всі оцінки порівнюються і обговорюються. Цей процес повторюється до тих пір, поки не буде досягнуто згоди щодо остаточного варіанту попередньої трудомісткості.

3. Оцінка за аналогією. Цей метод використовується в тому випадку, якщо в даній області застосування створюваного програмного проєкту вже реалізовані аналогічні проєкти. Метод заснований на порівнянні запланованого проєкту з попередніми проєктами, що мають подібні характеристики. Метод використовує експертні дані або збережені дані про проєкт. Експерти обчислюють високу, низьку і найбільш ймовірну оцінку трудомісткості, ґрунтуючись на відмінностях між новим проєктом і попередніми проєктами. Оцінка може бути досить детальною залежно від глибини аналогій. Слабкість моделі полягає в тому, що ступінь подібності нового проєкту і попередніх проєктів, як правило, не дуже велика.

Найкращий варіант – це використання накопичених в організації історичних даних, що дозволяють зіставити трудомісткість вашого проєкту з трудомісткістю попередніх проєктів аналогічного розміру. Однак це можливо тільки за таких умов:

- в організації акуратно документуються реальні результати попередніх проєктів;
- принаймні один з попередніх проєктів (а краще декілька) має аналогічний характер і розмір;
- життєвий цикл, обрані методи і засоби розробки, кваліфікація і досвід проєктної команди нового проєкту також подібні до тих, які мали місце в попередніх проєктах.

4. Закон Паркінсона. Згідно з цим законом зусилля, витрачені на роботу, розподіляються рівномірно за виділеним на проєкт часом. Критерієм для оцінки витрат за проєктом є людські ресурси, а не цільова оцінка самого програмного проєкту. Якщо проєкт, над яким працюють п'ять чоловік, повинен бути закінчений протягом 12 місяців, то витрати на його виконання обчислюються в 60 люд.-міс.

5. Оцінка з метою виграти контракт. Витрати на проєкт визначаються наявністю тих коштів, які є у замовника, тому трудомісткість проєкту залежить від бюджету замовника, а не від функціональних характеристик створюваного проєкту. Вимоги доводиться змінити так, щоб не виходити за межі прийнятого бюджету.

Кожен метод оцінки має слабкі і сильні сторони. Для роботи над великими проєктами необхідно застосувати декілька методів оцінки для їх подальшого порівняння. Якщо при цьому виходять абсолютно різні результати, інформації для отримання більш точної оцінки недостатньо. В цьому випадку необхідно скористатися додатковою інформацією, після чого повторити оцінку, і так до тих пір, поки результати різних методів не стануть досить близькими.

Описані методи оцінки можуть бути застосовні, якщо задокументовані вимоги майбутньої системи.

Найбільш впливовий чинник оцінки трудомісткості в цих моделях – розмір програмного проєкту. Процедура оцінки трудомісткості розробки програмного проєкту складається з наступних дій:

- оцінка розміру розроблювального проєкту;
- оцінка трудомісткості в людино-годинах (чол.-г.);
- оцінка тривалості проєкту в календарних місяцях;
- оцінка вартості проєкту.

Оцінка розміру проєкту базується на знанні вимог до системи. Для такої оцінки існують два основних способи:

1. За аналогією. Якщо в минулому доводилося мати справу з подібним проєктом і його оцінки відомі, то можна, ґрунтуючись на них, приблизно оцінити свій проєкт.

2. Шляхом підрахунку розміру за певними алгоритмами на основі вихідних даних – вимог до системи.

Проблеми оцінки розміру програмного проєкту:

- проблема може бути недостатньо добре зрозуміла для розродників і (або) замовників через те, що деякі факти були упущені або перекручені;
- недолік або повна відсутність історичних даних не дозволяє створити базу для оцінок в майбутньому;
- проєктуюча організація не має стандартів, за допомогою яких можна виконувати процес оцінювання (або в разі наявності стандартів їх ніхто не дотримується); в результаті спостерігається нестача сумісності при виконанні процесу оцінювання;
- менеджери проєктів вважають, що було б непогано фіксувати вимоги на початку проєкту, замовники ж вважають, що не варто витратити час на розробку специфікації вимог;

- помилки, як правило, ховаються, замість того, щоб оцінюватися і відображатися, в результаті чого створюється помилкове враження про фактичну продуктивність;

- менеджери, аналітики, розробники, тестувальники і ті, хто впроваджує проєкт, можуть мати різне уявлення про процеси оцінювання та про можливості вдосконалення проєкту.

Тема 4. ПЛАНУВАННЯ РОЗРОБКИ ПРОГРАМНИХ ПРОДУКТІВ.

4.1 Метод сітьового планування програмної діяльності

4.2 Застосування PERT-аналізу у плануванні проєктів

4.1 Метод сітьового планування програмної діяльності

Елементи сітьового графіку

Сітьова модель виконується у вигляді графіка, який складається зі стрілок та кружечків. В основі побудови сітьового графіку лежать поняття “робота” та “подія.”

Робота (\rightarrow) – виробничий процес, який вимагає певних затрат ресурсів та часу. Над стрілкою записують її тривалість у прийнятому масштабі часу (дні, місяці).

Фіктивна робота (залежність) (\rightarrow) – процес, який не вимагає затрат а ні ресурсів, а ні часу, але відображає взаємозв’язок між подіями. Залежність може бути технологічною, організаційною, умовною.

Подія (\bigcirc) – факт закінчення однієї або декількох робіт, необхідних та достатніх для початку наступних робіт.

Шлях (\longrightarrow) – безперервна послідовність робіт від вихідної до кінцевої події. Шлях, що має найбільшу тривалість, є критичним.

Всі події на сітковому графіку кодуються натуральним рядом чисел, без пропусків. Кожній події присвоюється самостійний номер. Стрілка

повинна бути направлена від події з меншим номером до події з більшим номером (рис. 4.1).

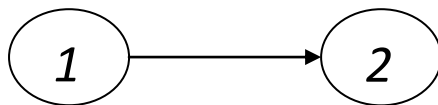


Рисунок 4.1 – Позначення події та роботи

Кожна робота сітьового графіку має часову оцінку – тривалість. Тривалість роботи може бути виміряна кількісно в одиницях часу (годинах, днях, тижнях, місяцях)

Події сітьового графіку можуть бути початковими та кінцевими (рис. 4.2).

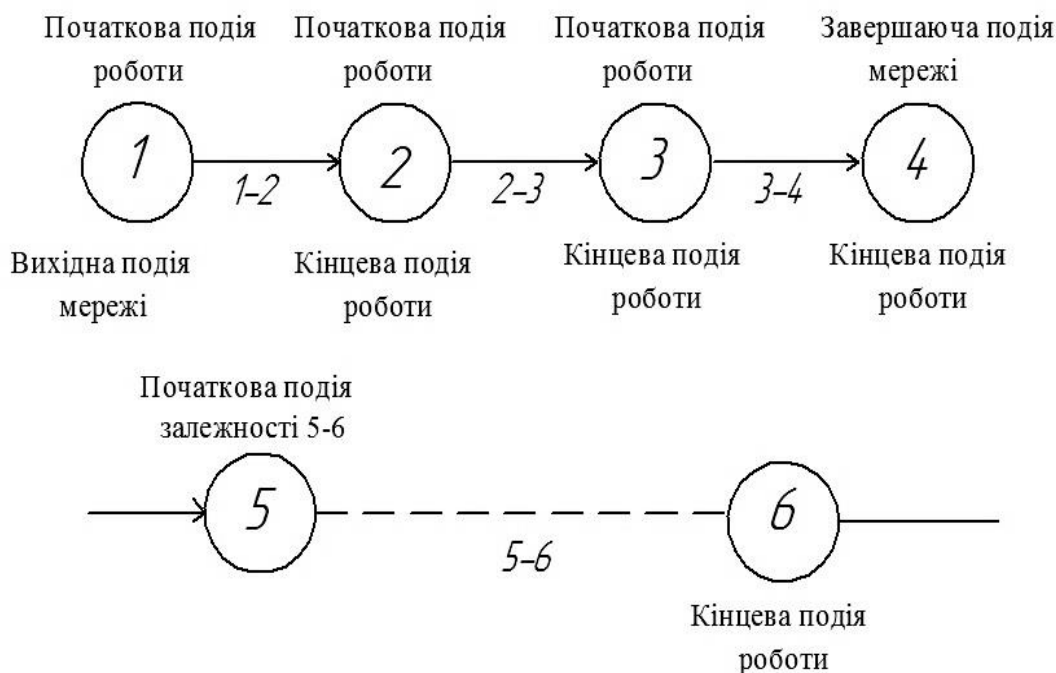


Рисунок 4.2 – Зображення подій, робіт та залежностей

Початкова подія визначає початок даної роботи і є кінцевою для попередніх робіт (рис. 4.2).

Кінцева подія визначає закінчення даної роботи і є початком для наступних робіт.

Вихідна подія – подія, яка не має попередніх робіт в межах сітьового графіку.

Завершаюча подія – подія, яка не має наступних робіт в межах сітьового графіку.

Складна подія – подія, в яку входить або виходить дві чи декілька робіт.

В сітьовому графіку між початковою та кінцевою подіями може бути декілька шляхів.

1. Шлях від початкової до кінцевої події сітьового графіку називають повним шляхом.

2. Шлях, що передує даній події, це шлях від початкової події до даної.

3. Шлях, що продовжується за даною подією – це шлях від даної події до кінцевої.

4. Шлях між подіями – це шлях, що з'єднує які-небудь дві події сітьового графіку, жодна з яких не є ні кінцевою, ні початковою.

Критичним шляхом називають повний шлях, який має найбільшу довжину із всіх повних шляхів. Його довжина визначає термін виконання робіт на сітьовому графіку.

В графіку може бути декілька критичних шляхів.

Роботи, які лежать на критичному шляху, називаються критичними.

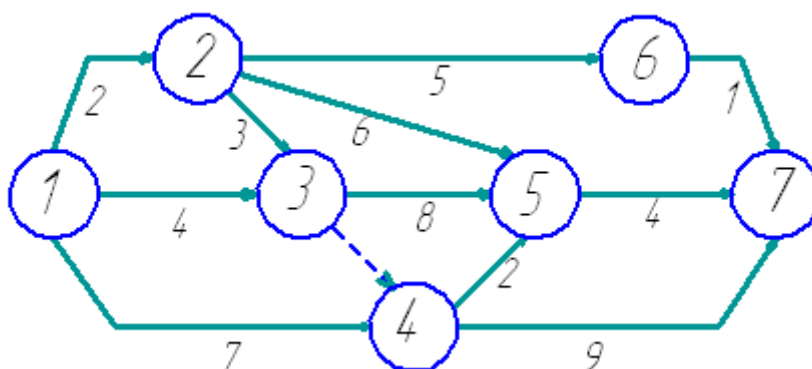


Рисунок 4.3 – Приклад сітьового графіку з часовими оцінками робіт

Тривалість шляху дорівнює сумі тривалостей робіт, що його складають

$$T(L_n) = t_{og} + \dots + t_{ij} + t_{jk} + \dots + t_{mn}.$$

Визначимо по табл. 4.1 усі шляхи від вихідної до завершального події сітьового графіку, приведеного на рис. 4.3.

Таблиця 4.1 – Шляхи сітьового графіку

Шлях (L_n)	Тривалість шляхів, дні $T(L_n)$
1, 2, 6, 7	$2+5+1=8$
1, 2, 5, 7	$2 + 6 + 4 = 12$
1, 2, 5, 6, 7	$2 + 6+1 + 1 = 10$
1, 2, 3, 5, 7	$2 + 3 + 8 + 4=17$
1, 2, 3, 5, 6, 7	$2+3+8+1+1= 15$
1, 3, 5, 7	$4 + 8 + 4=16$
1, 3, 5, 6, 7	$4 + 8 + 1 + 1=14$
1, 4, 5, 7	$7 + 2 + 4=13$
1, 4, 5, 6, 7	$7 + 2+1 + 1 = 11$
1, 4, 7	$7 + 9 = 16$

Усі шляхи сітьового графіку мають визначену тривалість. Повний шлях максимальної тривалості є критичним шляхом сітьового графіку ($T_{кр}$). У розглянутому прикладі критичним є шлях , 1, 2, 3, 5, 7, що має тривалість 17 одиниць часу.

Правила побудови сітьових графіків

При побудові сітьового графіку рекомендується направляти стрілки робіт з ліва на право і зображати їх (по можливості) горизонтальними лініями без перетинань.

У сітковому графіку необхідно дотримуватися умови, що кожна наступна робота може бути почата не раніше, ніж закінчаться всі попередні їй роботи, тобто для всіх пар залежних робіт, таких, як (i, j) передуює (k, l) , повинне дотримуватися нерівність (4.1):

$$t_{(kl)}^{(H)} \geq t_{ij}^{(O)}, \quad (4.1)$$

де $t_{kl}^{(H)}$ – початок роботи (k, l) ;

t – закінчення роботи (i, j) .

Крім того, для правильного відображення взаємозв'язків між роботами в сітьовому графіку при його побудові необхідно дотримуватися наступних **правил**.

1. Жодна з робіт не повинна мати однакового коду з іншою.
2. Не повинно бути замкнутих контурів (циклів).
3. Не повинне бути “тупиків”, тобто подій, з яких не виходить ні однієї роботи, і “хвостів”, тобто подій, у котрі не входить ні однієї роботи.
4. Між двома подіями більше, ніж одна робота бути не може. Якщо необхідно вказати логічну послідовність двох робіт, вводиться фіктивна робота, яка не має тривалості та позначається пунктирною лінією.
5. Нумерація подій мусить бути такою, щоб номер початкової події був менший номера кінцевої події.

Графічний метод розрахунку параметрів сітьового графіку

Умовні позначення прийняті такі:

$i - j$ – код даної роботи;

i – код початкової події даної роботи;

j – код кінцевої події даної роботи;

$h - i$ – код робіт, що передують даній роботі;

h – код подій, що передують початковій події даної роботи;

$j - k$ – код робіт, що впливають за кінцевою подією даної роботи;

k – код подій, що впливають за кінцевою подією даної роботи;

L – шлях;

$L_{кр.}$ – критичний шлях;

t – тривалість шляху;

$T_{кр.}$ – тривалість критичного шляху;

t_{i-j} – тривалість роботи;

$T_{р.п}$ – ранній початок роботи;

$T_{р.з}$ – раннє закінчення роботи;

T_p – ранній термін події, що відбулася i ;

$T_{п.п}$ – пізній початок роботи $i-j$;

$T_{п.з}$ – пізнє закінчення роботи $i-j$;

$T_п$ – пізній термін події, що відбулася j ;

R_{i-j} – загальний або повний резерв часу роботи $i-j$;

r_{i-j} – частковий резерв часу роботи $i-j$.

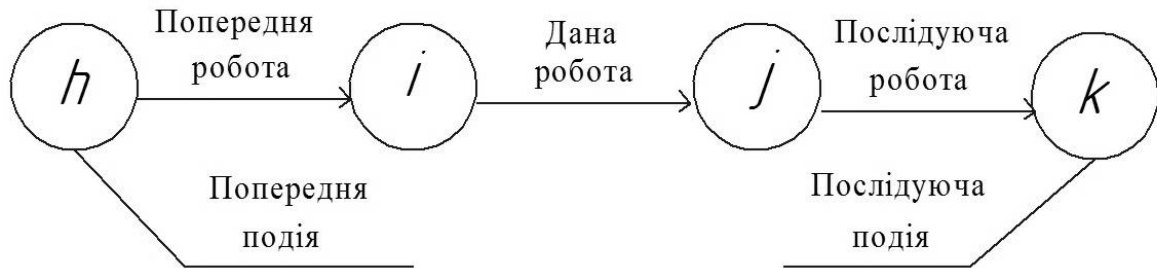


Рисунок 4.4 – Схема кодування робіт та подій на сітковому графіку

Часові параметри. Кожна робота сітьового графіку має часову оцінку – тривалість. Тривалість роботи можна виміряти кількісно в одиницях часу (роках, місяцях, тижнях, днях, годинах). Тривалість роботи проставляють під (або над) стрілкою, що зображає цю роботу. Розрахунок параметрів сітьового графіку може виконуватись по роботах або по подіях.

Ранній строк початку роботи – мінімальний із можливих моментів початку певної роботи при заданих тривалостях робіт і початковому моменті. Він дорівнює найбільшому з ранніх строків закінчення безпосередньо попередніх робіт чи ранньому строку настання початкової події певної роботи:

$$t_{ij}^{p.п} = \max_{(vi \in i \in) c} t_{vi}^{p.з} = T^{(p)}. \quad (4.2)$$

Ранній строк закінчення роботи $t_{ij}^{(p.з.)}$ – мінімальний з можливих моментів закінчення певної роботи при заданих тривалостях робіт і початковому моменті.

Ранній строк закінчення певної роботи дорівнює сумі її раннього строку початку і тривалості:

$$t_{ij}^{(p.з.)} = t_{ij}^{(p.п)} + t_{ij}. \quad (4.3)$$

Для вихідних робіт сітьового графіку ранній строк початку дорівнює заданому моменту початку виконання робіт чи він дорівнює нулю, якщо момент початку не задано:

$$t_{ij}^{(p.п)} = T_0. \quad (4.4)$$

чи $t_{ij}^{(p.п)} = 0$, якщо T_0 не задано.

Ранній строк закінчення дорівнює сумі заданого моменту (T_0) виконання робіт і тривалості робіт, а за відсутності заданого моменту він дорівнює тривалості цих робіт:

$$t_{ij}^{(p.з.)} = T_0 + t_{ij} \quad (4.5)$$

чи $t_{ij}^{(p.з.)} = t_{ij}$.

Тривалість критичного шляху дорівнює найбільшому з ранніх строків закінчення завершальних робіт чи ранньому настанню завершальної події сітьового графіка:

$$T_{кр} = \max_{(vi \in c(iw))} t_{iw}^{п.з.} = T_{iw}^{п.} \quad (4.6)$$

Ранні строки настання подій. Ранній строк настання подій $T_i^{(p)}$ – максимальний із можливих моментів настання певної події при заданих тривалостях робіт і початковому моменті.

Ранній строк настання вихідної події сіткового графіка чисельно дорівнює величині заданого початкового моменту:

$$T_i^{п} = T_0.$$

Ранній строк настання будь якої події дорівнює найбільшій із сум ранніх строків настання початкових подій і тривалості робіт, що входять до події, що розглядається:

$$T_j^{п} = \max_{j \in B(j)} T_i^{п} + t_{ij}. \quad (4.7)$$

Пізній строк закінчення роботи $t_{ij}^{п.з.}$ – максимальний із допустимих моментів закінчення цієї роботи, при якому ще можливе виконання всіх наступних робіт у межах критичного шляху.

Пізній строк закінчення цієї роботи дорівнює найменшому з пізніх строків початку безпосередньо наступних робіт чи пізньому строку настання кінцевої події цієї роботи:

$$t_{ij}^{п.з.} = \min_{(ij) \in c(j)} t_{ik}^{п.п.} = T_j^{п.} \quad (4.8)$$

Пізній строк початку роботи $t_{ij}^{п.п.}$ – максимальний із допустимих моментів початку цієї роботи, при якому ще можливе виконання всіх наступних робіт у межах критичного шляху.

Пізній строк початку цієї роботи дорівнює різниці між величиною її пізнього строку закінчення і тривалості:

$$t_{ij}^{п.п.} = t_{ij}^{п.з.} - t_{ij}. \quad (4.9)$$

Для завершальних робіт сітьового графіку: пізній строк закінчення дорівнює величині тривалості критичного шляху:

$$T_{iw}^{п.з.} = T_{кр} = \max t_{iw}^{п.з.} \quad (4.10)$$

Пізній строк початку завершальної роботи дорівнює різниці між тривалістю критичного шляху і тривалістю цієї роботи:

$$T_{iw}^{п.з.} = T_{кр} - t_{iw} \quad (4.11)$$

Пізній строк настання події $T_i^{(п)}$ – максимальний із допустимих моментів настання цієї події, при якому ще можливе виконання всіх наступних робіт у межах критичного шляху.

Пізній строк настання завершальної події сітьового графіка чисельно дорівнює ранньому строку його настання:

$$T_{ij}^{п.з.} = T_{iw}^{п.з.} \quad (4.12)$$

Пізній строк настання будь-якої події дорівнює найменшій різниці між пізнім строком настання кінцевої події і тривалості робіт, які виходять із події, що розглядається:

$$T_i^{п.з.} = \min_{(j) \in (i)} [T_j^{п.з.} + t_{ij}] \quad (4.13)$$

Повний резерв $[r_{ij}^{п.з.}]$ – максимальний час, на який можна відстрочити початок чи збільшити тривалість роботи (ij) , не змінюючи ранній строк настання завершальної події сітьового графіка.

Повний резерв часу роботи дорівнює резерву часу найбільшого за тривалістю шляху, що проходить через цю роботу, тому використання його повністю на одній роботі анулює повні резерви часу всіх робіт, що лежать на цьому шляху. Величина повного резерву визначається різницею пізнього і раннього строків початку чи пізнього і раннього строків закінчення цієї роботи:

$$r_{ij}^n = t_{ij}^{n.p.} - t_{ij}^{p.p.}, \quad (4.14)$$

чи

$$r_{ij}^n = t_{ij}^{n.z.} - t_{ij}^{p.z.}. \quad (4.15)$$

Коли визначені строки настання подій, повний резерв часу роботи визначається як різниця пізнього строку настання кінцевої події і суми раннього строку настання початкової події і тривалості цієї роботи:

$$r_{ij}^n = [T_j^n + t_{ij}]. \quad (4.16)$$

Вільний резерв часу $[r_{ij}^B]$ – максимальний час, на який можливо відкласти початок чи збільшити тривалість роботи (ij) за умови, що всі події сітки настають у свої ранні строки.

Використання вільного резерву на одній із робіт не змінює величини вільних резервів часу решти робіт сітки.

Вільний резерв часу роботи визначається як різниця між раннім строком початку безпосередньо наступних робіт і раннім строком закінчення даної роботи:

$$r_{ij}^B = t_{jk}^{p.p.} - t_{ij}^{p.z.}. \quad (4.17)$$

Коли визначено строки настання події, вільний резерв часу роботи визначається як різниця раннього строку настання кінцевої події і суми раннього настання події і тривалості цієї роботи (ij):

$$r_{ij}^B = T_i^p - [T_j^p + t_{ij}]. \quad (4.18)$$

Вільні резерви часу мають місце, коли однією подією закінчується не менше двох робіт.

Резерв часу настання події показує, на який гранично допустимий строк можливо затримати настання цієї події, не спричиняючи при цьому

збільшення строку виконання проєкту, і визначається як різниця між пізнім і раннім строками настання даної події:

$$R_i = T_i^{\text{п}} - T_i^{\text{р}} \quad (4.19)$$

Резерв часу робіт і подій, що складають критичний шлях, дорівнюють нулю.

Алгоритм розрахунку сітьового графіку в табличній формі

Для розрахунку сітьового графіку в таблиці необхідно, щоб події були пронумеровані таким чином: номер початкової події кожної роботи повинен бути менше номера її кінцевої події. Вихідній події привласнюється перший номер, а всі наступні події одержують номери в порядку зростання від початкового до завершального. Після нумерації кожна робота одержує свій код, відповідний номерам її початкової і кінцевої подій.

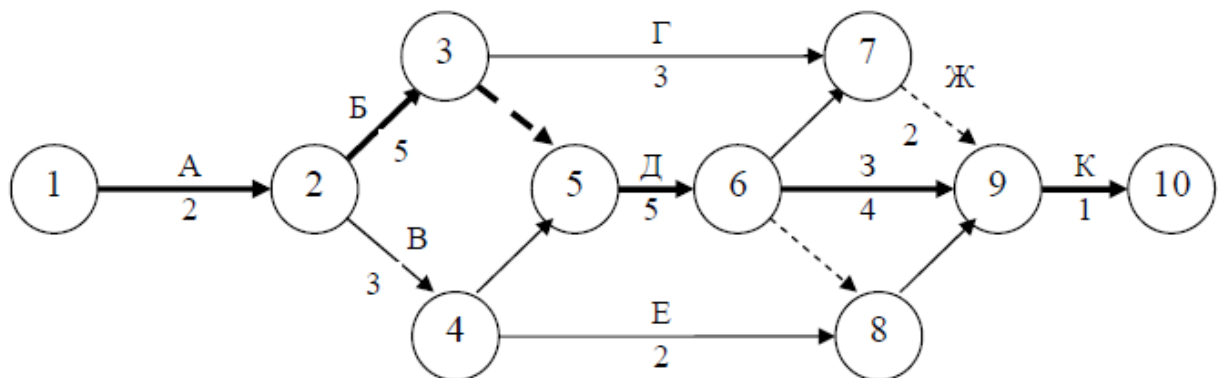


Рисунок 4.5 – Розрахунок сітьового графіка

Початкові дані з графіка для розрахунку заносять в графи 1, 2 і 3 таблиці 4.2. Всі ці три графи заповнюють одночасно.

У графу 1 заносять номери початкових подій попередніх робіт. Наприклад, для роботи (7,9) передуючими є робота (3,7) і залежність (6,7), отже в гр.1 заносять номери початкових події цих робіт 3 і 6.

У графу 2 заносять коди робіт і залежностей в порядку зростання початкових номерів подій, тобто спочатку роботи, що виходять з події 1, потім з події 2 і т.д.

У графу 3 проставляються тривалості робіт.

Ранні терміни початку і закінчення робіт розраховують по таблиці 2 зверху вниз. Ранній початок робіт, що виходять з першої події, рівний нулю. Раннє закінчення – сумі раннього початку і тривалості роботи:

$$t_{i,1}^{p,3} = t_{i,1}^{p,n} + t_{1,1}.$$

Таблиця 4.2 – Розрахунок сітьового графіку

Номери початкових подій попередніх робіт	Код робіт	Тривалості робіт	Ранній початок робіт $t_{i,j}^{pn}$	Раннє закінчення робіт $t_{i,j}^{pz}$	Пізній початок робіт $t_{i,j}^{nn}$	Пізнє закінчення робіт $t_{i,j}^{nz}$	Загальний резерв часу $R_{i,j}$	Частковий резерв часу $r_{i,j}$
1	2	3	4	5	6	7	8	9
–	(1.2)	2	0	2	0	2	0	0
1	(2.3)	5	2	7	2	7	0	0
1	(2.4)	3	2	5	4	7	2	0
2	(3.5)	0	7	7	7	7	0	0
2	(3.7)	3	7	10	14	17	7	2
2	(4.5)	0	5	5	7	7	2	2
2	(4.8)	2	5	7	10	12	5	5
3.4	(5.6)	5	7	12	7	12	0	0
5	(6.7)	0	12	12	17	17	5	0
5	(6.8)	0	12	12	12	12	0	0
5	(6.9)	4	12	16	15	19	3	3
3.6	(7.9)	2	12	14	17	19	5	5
4.6	(8.9)	7	12	19	12	19	0	0
6.7.8	(9.10)	1	19	20	19	20	0	0

Наприклад, для роботи (1,2):

$$t_{1,2}^{p,3} = t_{1,2}^{p,\Pi} + t_{1,2} = 0 + 2 = 2.$$

Ранній початок подальших робіт рівний максимальному з ранніх закінчень попередніх робіт:

$$t_{i,k}^{p,\Pi} = \max\{t_{i,l}^{p,3}\}.$$

Наприклад, для роботи (5,6):

$$t_{5,6}^{p,\Pi} = \max\{t_{3,5}^{p,3}; t_{4,5}^{p,3}\} = \max\{7; 5\} = 7.$$

Так само визначають ранні початки і закінчення всіх робіт і заносять в **графи 4 і 5** табл.

Максимальне раннє закінчення робіт, що входять в завершальну подію, визначає тривалість критичного шляху. В даному прикладі $T_{кр.} = 20$.

Пізні терміни початку і закінчення робіт записують в **графи 6 і 7** табл.

Розрахунок ведуть в таблиці від низу до верху.

Для робіт, що входять в завершальну подію, пізні закінчення дорівнює тривалості критичного шляху: $t_{9,10}^{п,3} = 20$.

Пізній початок будь-якої роботи визначається різницею між її пізнім закінченням і тривалістю:

$$t_{i,j}^{п,\Pi} = t_{i,j}^{п,3} - t_{i,j}.$$

Наприклад, для роботи (9,10):

$$t_{9,10}^{п,\Pi} = t_{9,10}^{п,3} - t_{9,10} = 20 - 1 = 19.$$

Пізні закінчення будь-якої роботи дорівнює найменшому пізньому початку подальших робіт:

$$t_{i,j}^{п,3} = \min t_{i,j}^{п,\Pi}.$$

Наприклад, для роботи (2,4): $t_{2,4}^{п,3} = \min\{t_{4,5}^{п,\Pi}; t_{4,5}^{п,\Pi}\} = \min\{7; 10\} = 7$.

Так само визначають пізні терміни всіх робіт сітьового графіку.

Повний резерв часу рівний різниці пізніх і ранніх термінів:

$$R_{i,j} = t_{i,j}^{p,3} - t_{i,j}^{p,3} = t_{i,j}^{p,p} - t_{i,j}^{p,p}.$$

Наприклад, для роботи (2,4): $R_{2,4} = t_{2,4}^{p,3} - t_{2,4}^{p,3} = 7 - 5 = 2 = t_{2,4}^{p,p} - t_{2,4}^{p,p} = 4 - 2 = 2$

Повний резерв часу заносять **в графу 8**.

У робіт критичного шляху повний резерв часу рівний нулю. Визначаємо критичні роботи, тобто роботи, що лежать на критичному шляху, це – 83 (1,2); (2,3); (3,5); (5,6); (6,8); (8,9); (9,10). Критичний шлях даного сітьового графіка буде (1, 2, 3, 5, 6, 8, 9, 10).

Вільний резерв часу заносять **в графу 9**, він визначається різницею між раннім початком подальшої роботи і раннім закінченням даної роботи:

$$r_{i,j} = t_{j,k}^{p,p} - t_{i,j}^{p,3}.$$

Наприклад, для роботи (3,7):

$$r_{3,7} = t_{7,9}^{p,p} - t_{3,7}^{p,3} = 12 - 10 = 2.$$

Вільний резерв часу роботи завжди менше або дорівнює її повному резерву:

$$r_{i,j} \leq R_{i,j}.$$

За вищеописаним алгоритмом всі розрахунки проводять в таблиці, використовуючи приведені формули.

Основними перевагами сітьового планування є:

1. Можна моделювати досить складні проекти, при цьому сітьові графіки залишаються компактними.
2. Забезпечується наочність проекту.
3. На сітьовому графіку можна виділити ланцюжок робіт, від яких залежить тривалість виконання всього комплексу.
4. Сітьовий графік проекту можна оптимізувати.
5. Сітьовий графік після оптимізації можна використати для управління ходом виконання проекту.

Разом з тим, ефективність використання сітьового планування в значній мірі залежить від:

- a) правильної структуризації робіт;

- б) реальної оцінки їхньої тривалості;
- в) професіоналізму розроблювачів;
- г) якості виконання робіт відповідно до розробленого сітьового графіку.

Алгоритм побудови сітьового графіку містить у собі:

- 1) зіставлення докладного переліку робіт, що входять у комплекс
- 2) визначення взаємозв'язку виконання робіт
- 3) оцінка тривалості виконання роботи
- 4) побудова сітьового графіка
- 5) визначення резервів часу й критичного шляху
- 6) оцінка всіх необхідних витрат
- 7) оптимізація сітьового графіку за обраним критерієм
- 8) здійснення управління проєктами на основі оптимізованого сітьового графіку.

4.2 Застосування PERT-аналізу у плануванні проєктів

PERT – це абревіатура від англійських Program Evaluation and Review Technique – технологія оцінки і перегляду програми, яка базується на ідеї сітьового планування. Відповідно методики PERT-аналізу проєкти розглядаються як мережі окремих подій та робіт, причому робота в таких мережах (сітях) являє собою будь-який елемент проєкту, на виконання якого необхідний час, і який може затримати початок виконання наступних робіт.

За розробленої моделі проєкту у вигляді сітьового графіка, в складі якого розгорнуто роботи і зв'язки між ними, будується схема. Для того, щоб розрахувати для кожної роботи дату початку і дату закінчення, потрібно оцінити тривалості всіх робіт з оцінкою того, скільки вони займуть часу. Тільки після цього РМ зможе цю модель розрахувати і зрозуміти терміни закінчення робіт. Метод PERT оперує трьома оцінками:

- оптимістичною;
- песимістичною;
- найбільш вірогідною.

Виходячи з отриманих відповідей із задалегідь заданою вірогідністю проводиться розрахунок відповідних термінів. При цьому рекомендується застосовувати техніку ковзаючого перегляду термінів з урахуванням досягнутих результатів.

Очікуваний час виконання роботи являє собою математичне очікування випадкової величини (у цьому випадку – тривалості робіт). Найбільше розповсюдження для визначення характеру розподілу тривалості робіт отримав закон β -розподілу. Тоді, очікуваний час виконання роботи можна визначити за формулою:

$$t_{\text{оч.}} = \frac{t_{\text{min}} + 4t_{\text{н.в.}} + t_{\text{max}}}{6}, \quad (4.20)$$

де t_{min} – оптимістична оцінка – мінімальна тривалість роботи; мінімально можливий час виконання даної роботи при найсприятливіших обставинах;

t_{max} – песимістична оцінка – максимальна тривалість роботи; максимально довгий час виконання даної роботи при найнесприятливішому збігу обставин;

$t_{\text{н.в.}}$ – найбільш вірогідна тривалість роботи; час виконання даної роботи при стандартних умовах.

За наявності лише двох оцінок (t_{min} та t_{max}) тривалість роботи обчислюється

$$t_{\text{оч.}} = \frac{3t_{\text{min}} + 2t_{\text{max}}}{5}. \quad (4.21)$$

Дисперсія – у даному випадку – міра розкиду очікуваного часу, дорівнює:

$$\delta^2_{t_{\text{оч.}}} = \frac{(t_{\text{max}} - t_{\text{min}})^2}{6}. \quad (4.22)$$

Метод усереднення тривалостей робіт, отриманих на основі імовірнісних оцінок, дозволяє імовірнісний сітьовий графік розглядати як детермінований.

Тема 5. ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

5.1 Критерії успішності проєкту

5.2 Прибуток та рентабельність розробки програмного продукту

5.3 Економічна ефективність капіталовкладень у програмний проєкт

5.4 Інвестиційна привабливість програмних проєктів

5.1 Критерії успішності проєкту

Завдання проєкту – досягнення конкретної бізнес-цілі, при дотриманні обмежень «залізного трикутника» (рис. 5.1). Це означає, що жоден з кутів трикутника не може бути змінений без здійснення впливу на інші. Наприклад, щоб зменшити час, потрібно буде збільшити вартість і / або скоротити зміст.

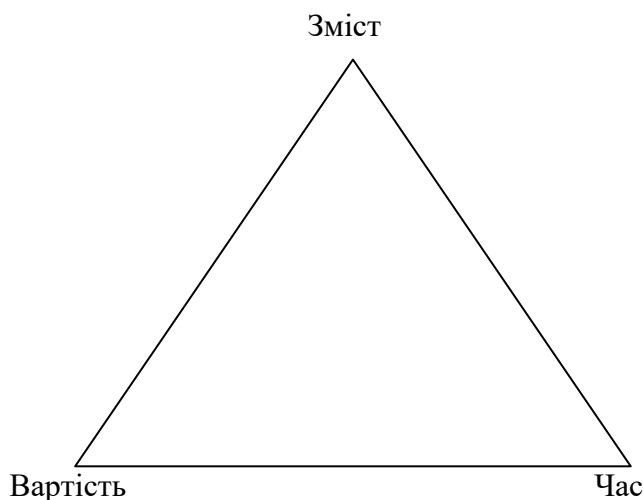


Рисунок 5.1 – «Залізний трикутник» обмежень проєкту

Згідно поточної редакції стандарту РМВОК, проєкт вважається успішним, якщо задоволені всі вимоги замовника та учасників проєкту. Тому у проєкті розробки програмного забезпечення сьогодні не три, а чотири фактори успіху:

1. Виконано у відповідності зі специфікаціями.
2. Виконано в строк.
3. Виконано в межах бюджету.
4. Кожен учасник команди йшов з роботи о 18:00 з почуттям успіху.

Цей четвертий фактор успіху повинен стати відтвореним, якщо підприємство хоче бути ефективним. Для успішного проєкту характерно постійне відчуття його учасниками почуття задоволення і гордості за результати своєї роботи, почуття оптимізму. Немає нічого більш згубного для проєкту, ніж байдужість або зневіра його учасників.

Ефективність – це відношення отриманого результату до необхідних витрат. Не можна розглядати ефективність, виходячи тільки з результативності: чим більше ти виробляєш, чим більше робиш, тим вище твоя ефективність. З таким підходом можна «зарізати на вечерю курку, яка несе золоті яйця». Витрати не слід плутати з інвестиціями. Оплата оренди, електроенергії, комунальні платежі – витрати. Створення і закріплення ефективної команди – це стратегічне придбання компанії. Навчання учасників проєкту – інвестиції. Вкладення в людей – це збільшення чисельника у формулі ефективності. Вихід з компанії всіх професіоналів після проєкту, виконаного за принципом «будь-якою ціною», – витрати, причому дуже важко поновлювані. Зростаюча конкуренція вказує на абсолютно чіткий тренд у світовій економіці – персонал – це форма інвестицій, активів, які потрібно вміти нарощувати, управляти і зберігати. Сьогодні люди – це капітал.

5.2 Прибуток та рентабельність розробки програмного продукту

Прибуток – найпростіша і одночасно найбільш складна категорія ринкової економіки. Він є основною і головною рушійною силою економіки ринкового типу, основним мотивом діяльності підприємств в цій економіці. **Прибуток** – це винагорода, яку одержує підприємець. Це є плата за те, що він організовує виробництво, управляє ним, впроваджує інновації (нововведення), ризикує. Виконуючи свої функції, підприємство забезпечує перевищення виручки (доходу) від реалізації продукції фірми над витратами на виробництво.

Прибуток, як основне джерело фінансування розвитку підприємства, вдосконалення його матеріально-технічної бази, забезпечення всіх форм інвестування, зумовлює необхідність спрямування всіх напрямів діяльності підприємства на принаймні збереження належного рівня прибутку або забезпечення його зростання. Прибуток, як частина виручки від реалізації продукції підприємства, з якої вилучено витрати виробничої та комерційної діяльності, відображає мету підприємницької діяльності та виявляється основним показником її ефективності.

Таким чином, прибуток як виражений у грошовій формі чистий дохід підприємства на вкладений капітал характеризує його винагороду за ризик здійснення підприємницької діяльності і є різницею між сукупним доходом і сукупними затратами в процесі здійснення цієї діяльності

Прибуток є основним джерелом фінансування розвитку підприємства, вдосконалення його матеріально-технічної бази та продукції, забезпечення всіх форм інвестування. Водночас це – об'єкт оподаткування, джерело сплати податків, а відтак – джерело розв'язання соціально-економічних цілей держави.

Прибуток розраховується за формулою

$$\Pi = \frac{C}{100\%} \cdot R, \quad (5.1)$$

де R –рівень рентабельності, %

Рентабельність – це загальний показник економічної ефективності роботи підприємства або використання капіталу і ресурсів.

Рентабельність проєкту – це фінансовий показник ефективності інвестицій в бізнес.

Для розрахунку рентабельності проєкту існує універсальна формула

$$P = V_{\pi} / V, \quad (5.2)$$

де V_{π} – вид прибутку;

V – виручка.

Стосовно до ІТ-проєкту змінна «вид прибутку» – це власне віддача від проєкту, а «показник, рентабельність якого потрібно розрахувати» – це вартість проєкту. Таким чином, формула отримує вигляд:

$$R = (\text{Віддача проєкту} / \text{Повна вартість проєкту}) \cdot 100\%.$$

Результат покаже суму доходу, який ви отримаєте після реалізації проєкту, на кожну грошову одиницю, вкладену в проєкт. Ці результати ви зможете порівняти з іншими інвест-проєктами з аналогічною сфери. Мінімального рівня рентабельності не існує, значення вважається позитивним, якщо воно вище депозитної ставки і альтернативних проєктів.

Депозитна ставка – відсоток від суми депозиту, який ви отримуєте за умови інвестиції.

5.3 Економічна ефективність капіталовкладень у програмний проєкт

Інвестиційні проєкти існують зазвичай у вигляді деякої сукупності альтернативних варіантів, реалізація яких пов'язана з певними фінансовими витратами. У зв'язку з цим у керівництва компанії виникають наступні питання:

– який обсяг інвестицій із загального фінансового плану вкладати в розвиток інфраструктури інформатизації;

– який з альтернативних варіантів створення (розвитку) технічного та програмного забезпечення вибрати. У будь-якому випадку витрати повинні бути економічно виправдані. За аналогією з тим, що поточні витрати в загальному випадку повинні покриватися поточними доходами (в іншому випадку ці витрати можуть вважатися економічно недоцільними), витрати капітального характеру повинні покриватися майбутніми доходами, а точніше майбутнім прибутком від комерційного застосування капіталовкладень. Отже, з економічної точки зору розгляд проєкту зі створення та розвитку інформаційних технологій в якості інвестиційного означає необхідність економічного обґрунтування необхідних витрат і оцінки ефективності передбачуваних інвестицій.

Своєчасний аналіз витрат і очікуваних результатів дозволяє менеджерам:

- порівнювати альтернативні варіанти реалізації ПП;
- отримувати співвідношення переваг одного програмного продукту над іншим при виборі ПП для впровадження;
- визначати економічні конкурентні переваги створеної програми над аналогічними продуктами конкурентів.

Існує ряд різних моделей і методів, використовуваних для порівняння аналізу витрат і очікуваних результатів. На практиці переважають дисконтні методи і відповідний їм набір показників:

– **чиста приведена вартість (NPV)** – дає змогу порівняти між собою теперішню вартість майбутніх доходів від реалізації інвестиційного проєкту (у фазі його експлуатації) з інвестиційними витратами, які необхідно здійснити в поточному періоді.;

– **внутрішня ставка прибутковості (IRR)** – ставка дисконтування, при якій чиста поточна вартість інвестиційного проєкту дорівнює нулю;

– **термін окупності проєкту** – час, який потрібен для того, щоб сума надходжень від реалізації проєкту відшкодувала суму витрат на його впровадження;

– **коефіцієнт прибутковості інвестицій (ROI)** – фінансовий коефіцієнт, який ілюструє рівень прибутковості або збитковості бізнесу, враховуючи суму зроблених в цей бізнес інвестицій;

– **сукупна вартість володіння TCO** – це загальна величина цільових витрат, які змушений нести власник з моменту вступу в стан володіння до моменту виходу зі стану володіння.

5.4 Інвестиційна привабливість програмних проєктів

Величину витрат і розміри інвестицій можна визначити на поточний момент часу, а очікуваний результат у вигляді прибутку з'явиться в майбутніх періодах. Оскільки гроші, отримані в майбутньому, будуть «коштувати» менше, ніж якби вони були отримані сьогодні, майбутній прибуток слід скоригувати з вартістю грошей в кожному з розглянутих інтервалів планування.

Інструментом приведення є норма дисконту E , яку називають також **нормою прибутковості або ставкою порівняння**, що є прийнятний для інвестора відсоток повернення на капітал, що інвестується за визначений період нарахування. Технічно приведення виконується шляхом множення обсягу платежів та надходжень в t -му періоді планування на коефіцієнт (відсоткову ставку) дисконтування α_t , що визначається для постійної норми дисконту E і t_0 за формулою

$$\alpha_t = 0; \alpha_t = \frac{1}{(1 + E)^t}, \quad (5.3)$$

де $0 \leq \alpha_t \leq 1, t = 1, 2, \dots, n$.

Чиста дисконтована (приведена) вартість NPV відноситься до показників ефективності капіталовкладень і характеризує чистий абсолютний результат або віддачу від реалізації інвестиційного проєкту:

$$NPV = \sum_{t=0}^T (R_t - Z_t) \cdot \alpha_t, \quad (5.4)$$

де R_t – загальний (валовий) результат експлуатації інвестицій, що досягається на t -му кроці розрахункового періоду;

Z_t – загальні витрати на t -му кроці розрахункового періоду;

T – розрахунковий період;

$(R_t - Z_t)$ – чистий економічний ефект E_t від використання інвестицій на t -му кроці.

При виділенні із загальних витрат обсягу капітальних вкладень (інвестиційних витрат) K_t і поточних витрат N_t формула для розрахунку NPV виглядає наступним чином:

$$NPV = \sum_{t=0}^T (D_t - K_t) \cdot \alpha_t, \quad (5.5)$$

де $D_t = R_t - N_t$.

Якщо значення $NPV > 0$, то даний проєкт можна вважати прибутковим, в іншому випадку він буде збитковим.

Внутрішня ставка (норма) дохідності (рентабельності) IRR показує, який відсоток прибутку необхідно закласти в розрахунки, щоб чиста теперішня вартість проєкту всіх майбутніх періодів дорівнювала нулю. Іншими словами, необхідно визначити значення дисконтної ставки, яка зрівнює наведену вартість майбутніх доходів і наведену вартість витрат. Якщо проєкт інвестується за рахунок банківського кредиту, то IRR визначає верхню межу допустимого рівня банківської процентної ставки, при якому очікуваний прибуток буде дорівнювати нулю:

$$\sum_{t=0}^T \frac{(D_t - K_t)}{(1 + E)^t} = 0. \quad (5.6)$$

Чисельне значення E можна визначити одним з наближених методів рішення рівнянь.

Термін окупності проекту PP визначається кількістю періодів, необхідних для повного повернення первісних інвестицій. Як правило, компанія, плануючи інвестиції в проекти, встановлює і мінімальний термін їх окупності, протягом якого сума дисконтованих чистих надходжень покриває суму наведених інвестиційних витрат, і в подальшому чистий інтегральний ефект залишається невід'ємним.

$$PP = t^* - \frac{NPV(t^*)}{NPV(t^*) - NPV(t^* - 1)}, \quad (5.7)$$

де $t^* = \min \{t\}$ – номер періоду, з якого чистий інтегральний ефект залишається невід'ємним (для ІТ-проектів рекомендований термін окупності не більше 3 років).

Коефіцієнт прибутковості інвестиції ROI дозволяє оцінити відносну ефективність вкладень в інформаційну структуру фірми і визначається за формулою

$$ROI = E_t / Z_t, \quad (5.8)$$

де E_t – економічний ефект від впровадження проекту;

Z_t – інвестиції (витрати) в проект.

Кожен з перерахованих вище параметрів не позбавлений недоліків. Так, чистий приведений дохід NPV описує в вартісному вираженні майбутній прибуток, однак не дозволяє безпосередньо порівнювати проекти з різним рівнем фінансування. Показник IRR усуває зазначений недолік, однак він не має зрозумілої економічної інтерпретації і не враховує обсяги початкових інвестицій. Термін окупності проекту не враховує зміну грошових потоків у часі.

Основною перевагою ROI є простота його визначення. Як фінансовий показник він зрозумілий менеджеру і дозволяє приводити до одного виміру і порівнювати різномасштабні проекти, легко порівнюється із загальною вартістю активів компанії. Однак розрахунки показників ґрунтуються на визначенні майбутнього прибутку від впровадження проекту, величина

якого не піддається точному визначенню. Незважаючи на математичну точність наведених формул, розрахунки носять приблизний характер і вимагають визначення деяких похибок (допусків).

Разом з тим зазначені недоліки не впливають на своєчасну оцінку інвестицій, що дозволяє керівникам бути не просто «прохачем» коштів, а виступати ініціатором ефективного інвестиційного проєкту, що конкурує на рівних з іншими інвестиційними пропозиціями щодо розвитку бізнесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архипенков С. Лекции по управлению программными проектами. Москва, 2009. – 127с.
2. Архипенков С. Руководство командой разработчиков программного обеспечения. Прикладные мысли. Москва, 2008. – 80с.
3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы: пер. с англ. Санкт-Петербург.: Символ-Плюс, 2006.
4. Демарко Т., Листер Т. Вальсируя с медведями. Управление рисками в проектах по разработке программного обеспечения. Компания p.m.Office, 2018. – 201с.
5. Дранишников Л. В. Конспект лекцій з дисципліни «Менеджмент проєктів програмного забезпечення». Кам'янське : ДДТУ, 2019. – 123 с.
6. Ехлаков Ю. П. Организация бизнеса на рынке программных продуктов: учебник. Томск : Изд-во Томск. гос. ун-та систем управления и радиоэлектроники, 2012. – 312 с.
7. Липаев В. В. Проблемы экономики производства крупных программных продуктов. Труды Института системного программирования РАН, 2008.
8. Липаев В. В. Техничко-экономическое обоснование проектов сложных программных средств. Москва : СИНТЕГ. 2004. – 284 стр.
9. Липаев В. В. Экономика программной инженерии заказных программных продуктов: учебное пособие. Москва: МАКС Пресс, 2014. – 148 с.
10. Липаев В. В. Экономика производства сложных программных продуктов. Москва : Синтег, 2008. – 432 с.
11. Макконнелл С. Остаться в живых. Руководство для менеджеров программных проєктов. Санкт-Петербург : «Питер», 2006.
12. Макконнелл С. Сколько стоит программный продукт. Санкт-Петербург: Питер, 2007. – 295с.

13. Ройс У. Управление проектами по созданию программного обеспечения: пер. с англ. Москва : ЛОРИ, 2002. – 424 с.

14. Сажин Ю. Б., Самохин С. В. Методическое пособие по выполнению организационно-экономической части дипломных проектов по разработке и использованию программных продуктов. Москва : МГТУ, 2004. – 42 с

15. Фатрелл Р. Т., Шафер Д. Ф., Шафер Л. И. Управление программными проектами. Достижение оптимального качества при минимуме затрат. Москва : Издательский дом “Вильямс”, 2004. – 1125 с.

ДЛЯ НОТАТОК

Навчальне видання

РОМАНЧИК Тетяна Володимирівна

ЕКОНОМІКА ТА ОРГАНІЗАЦІЯ ВИРОБНИЦТВА ПРОГРАМНИХ
ПРОДУКТІВ

Конспект лекцій
для студентів технічних спеціальностей,
усіх форм навчання

Відповідальний за випуск проф. Сисоєв В.В.
Роботу до видання рекомендував В.А. Кучинський

В авторській редакції

План 2021 р., поз. 49

Підписано 09.12.2021.
Гарнітура Times New Roman. Ум. друк. арк.3,7.

Видавець Видавничий центр НТУ «ХП».
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, Харків, вул. Кирпичова, 2
