Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

____Дослідження бізнес-моделей монетизації ІТ-проєктів з відкритим кодом__
(тема)

Виконав:
студент _2_ курсу, групи _УПГІТм-21-1____
__Владислав ШЕВЧЕНКО_____
(власне ім'я, прізвище)

Спеціальність _122 Комп'ютерні науки_____
(код і повна назва спеціальності)
Тип програми __освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)
Освітня програма _Управління проєктами в галузі інформаційних технологій_____
(повна назва освітньої програми)
Керівник _проф. каф. ІУС, Максим ЄВЛАНОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ _Костянтин ПЕТРОВ__
(підпис) (власне ім'я, прізвище)

2023 р.

Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____

Кафедра _____Інформаційних управляючих систем_____

Рівень вищої освіти _____другий (магістерський)_____

Спеціальність _____122 Комп'ютерні науки_____
(код і повна назва)

Тип програми _____освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)

Освітня програма_Управління проєктами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_03_» _квітня_ 20 23_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Шевченку Владиславу Олександровичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи _____Дослідження бізнес-моделей монетизації_____
_____ІТ-проєктів з відкритим кодом_____
_____

затверджена наказом університету від _03_ _квітня_2023_ р. № _319Ст_

2. Термін подання студентом роботи до екзаменаційної комісії _18_ _травня_ 2023 р

3. Вихідні дані до роботи сучасні методи планування ІТ-проєктів з відкритим кодом, бізнес-моделі монетизації ІТ-проєктів, особливості розповсюдження ІТ-проєктів з відкритим кодом, економічні моделі розрахунку прибутку від ІТ-проєктів, звітність компаній що працюють за стратегією відкритого коду _____
_____

4. Перелік питань, що потрібно опрацювати в роботі проаналізувати існуючі бізнес-моделі монетизації ІТ-проєктів з відкритим кодом, вибрати моделі для детального дослідження, теоретично описати опрацьовані моделі, практично застосувати описані моделі, проаналізувати отримані результати практичного застосування досліджуваних моделей_____
_____
_____
_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.04.2023	Виконано
2	Аналіз предметної галузі та постановка задачі	04.04.2023-06.04.2023	Виконано
3	Аналіз існуючих моделей монетизації ІТ-проєктів	06.04.2023-10.04.2023	Виконано
4	Дослідження моделі отримання прибутку від консультаційних послуг	10.04.2023-15.04.2023	Виконано
5	Практичне застосування моделі отримання прибутку від консультаційних послуг	15.04.2023-20.04.2023	Виконано
6	Дослідження моделі отримання прибутку від розробки відкритих платформ	20.04.2023-01.05.2023	Виконано
7	Практичне застосування моделі отримання прибутку від розробки відкритих платформ	01.05.2023-07.05.2023	Виконано
8	Підготовка пояснювальної записки та графічного матеріалу	07.05.2023-12.05.2023	Виконано
9	Надання пояснювальної записки на перевірку керівнику	12.05.2023-17.05.2023	Виконано
10	Захист перед ЕК	19.05.2023	Виконано

Дата видачі завдання _03_ __квітня__2023 р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Максим ЄВЛАНОВ
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка містить: 98 сторінок, 16 рисунків, 3 таблиці, 32 джерела, 7 додатків.

ВІДКИРИЙ КОД, МОНЕТИЗАЦІЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПРОПРІЄТАРНІСТЬ, РОЗПОДІЛ ПРИБУТКУ

Об’єктом дослідження є монетизація ІТ-проєктів з відкритим кодом.

Предметом дослідження є бізнес-моделі монетизації ІТ-проєктів з відкритим кодом.

Метою даної роботи є дослідження методів та особливостей монетизації ІТ-проєктів з відкритим кодом які використовуються у прогнозуванні прибутковості проєктів, застосуванні різних методів монетизації та їх реалізації задля використання у різних етапах планування та розробки ІТ-проєктів.

Методи дослідження – моделювання бізнес-процесів фірм розробників програмного забезпечення. В роботі представлено моделі, що описують фірми з різними підходами до монетизації. Перша модель описує фірму розробника відкритого ПЗ, що заробляє на продажі додаткових послуг. Наступна наведена модель описує розподіл прибутку між розробником платформ з відкритим кодом і фірмами користувачами платформою.

Використання наведених моделей дозволяє вносити зміни в планування та управлінням розробки ІТ-проєктів з відкритим кодом, щоб максимізувати прибуток, враховуючи всі обмеження відкритих проєктів.

ABSTRACT

The explanatory note to the qualification work contains: 98 pages, 16 figures, 3 tables, 32 sources, 7 appendixes.

SOFTWARE, OPEN SOURCE, MONETIZATION, PROFIT SHARING, PROPRIETARY NATURE

The object of research is the monetization of open source IT projects.

The subject of the study is business models for monetizing open source IT projects.

The purpose of this work is to study the methods and features of monetization of open source IT projects used in forecasting project profitability, applying various monetization methods and their implementation for use in various stages of planning and development of IT projects.

Research methods - modeling of business processes of software development firms. The paper presents models that describe firms with different approaches to monetization. The first model describes an open source software development firm that earns money by selling additional services. The next model describes the distribution of profits between the developer of open source platforms and firms using the platform.

Using these models, it is possible to make changes in the planning and management of open source IT project development to maximize profits, taking into account all the limitations of open source projects.

ЗМІСТ

Скорочення та умовні познаки 8
Вступ 9
1 Аналіз предметної області та постановка задачі 11
1.1 Аналіз особливостей розробки ІТ-проєктів відкритим кодом 11
1.2 Постановка задачі 14
2 Аналіз методів монетизації іт-проєктів з відкритим кодом 15
2.1 Аналіз методу подвійного ліцензування 15
2.2 Аналіз методу розробки комерційних розширень 18
2.3 Аналіз моделі монетизації послуг та підтримки 20
2.4 Аналіз моделі розробки платформ 21
2.5 Порівняння розглянутих моделей монетизації 22
3 Дослідження моделі отримання прибутку від консультаційних послуг 24
3.1 Опис моделі монетизації ІТ-проєктів шляхом продажу продукту та послуг 26
3.2 Практичне застосування розробленої моделі 33
3.2.1 Характеристика ринку споживачів відкритого програмного забезпечення 33
3.2.2 Визначення стратегії розповсюдження в залежності від вартості розробки 36
4 Дослідження моделі отримання прибутку від розробки відкритих платформ 41
4.1 Опис моделі монетизації платформи з відкритим кодом 41
4.1.1. Розміри мереж користувачів та компонентів 42
4.1.2 Прибутки від продажу додатків 44
4.1.3. Прибуток від реклами в безкоштовних додатках 46
4.2 Практичне застосування моделі монетизації розробки платформ 49

4.2.1 Визначення ролі розміру мережі користувачів на прибуток від продажу 49
4.2.2 Максимізація прибутку за рахунок корегування мережі користувачів 50
4.2.3 Визначення оптимального розподілу прибутку від продажів додатків та реклами 51
4.2.4 Візуалізація розподілу прибутку та розміру мережі користувачів... 52
Висновки 57
Перелік джерел посиланя 58
Додаток А Доказ залежності ринку від якості послуг....**Ошибка! Закладка не определена.**
Додаток Б Доказ залежності ринку від зусиль на розробку та ціни продукту ..... **Ошибка! Закладка не определена.**
Додаток В Доказ впливу вартості розробки на вибір стратегії розповсюдження ..... **Ошибка! Закладка не определена.**
Додаток Г Доказ впливу мережі користувачів на прибутки від продажу додатків ..... **Ошибка! Закладка не определена.**
Додаток Д Доказ впливу мережі користувачів на прибутки від продажу реклами ..... **Ошибка! Закладка не определена.**
Додаток Е Доказ знаходження оптимального розподілу прибутку..... **Ошибка! Закладка не определена.**
Додаток Ж Графічний матеріал ..... **Ошибка! Закладка не определена.**

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ВПЗ – вільне програмне забезпечення;

ІТ – інформаційні технології;

ПЗ – програмне забезпечення;

ПЗВВК – програмне забезпечення з відкритим вихідним кодом;

BSD – Berkeley Software Distribution;

CRM – customer relationship management;

ERP – enterprise resource planning;

GPL – General Public License;

OSS – open-source software.

ВСТУП

Програмне забезпечення з відкритим кодом стало фундаментальною частиною сучасної IT-інфраструктури, оскільки багато підприємств та організацій покладаються на нього для виконання критично важливих операцій. Філософія відкритого коду базується на ідеї, що програмне забезпечення повинно бути вільно доступним, модифікованим і розповсюджуваним. Цей підхід призвів до створення величезної екосистеми проєктів програмного забезпечення з відкритим кодом, починаючи від невеликих утиліт і закінчуючи великими системами корпоративного рівня.

Незважаючи на широке розповсюдження, багато проєктів з відкритим кодом намагаються знайти стійкі бізнес-моделі, які дозволять їм продовжувати розробку і підтримку в довгостроковій перспективі. У той час як деякі проєкти підтримуються великими корпораціями, багато інших розробляються і підтримуються спільнотами волонтерів або невеликими командами розробників. Монетизація програмного забезпечення з відкритим кодом є складним питанням, яке вимагає ретельного розгляду різноманітних факторів, включаючи ліцензування, розповсюдження та залучення спільноти.

Монетизація програмного забезпечення з відкритим кодом є актуальним предметом дослідження оскільки розробникам потрібно отримувати дохід, щоб підтримувати постійну розробку та підтримку свого програмного забезпечення, в той же час коли багато розробників і користувачів програмного забезпечення з відкритим кодом вважають, що програмне забезпечення має бути безкоштовним, як з точки зору вартості, так і з точки зору його доступності для модифікації та розповсюдження.

Ця робота має на меті вивчити бізнес-моделі монетизації IT-проєктів з відкритим кодом. Основна мета - вивчити різні стратегії та підходи, які використовують проєкти з відкритим кодом для отримання доходу, зберігаючи при цьому дух філософії відкритого коду. Дослідження також

розгляне проблеми, з якими стикаються проєкти з відкритим вихідним кодом при монетизації свого програмного забезпечення, включаючи конкуренцію з пропрієтарним програмним забезпеченням, питання ліцензування та очікування спільноти.

Завдяки розумінню різних бізнес-моделей монетизації, що використовуються в IT-проєктах з відкритим кодом, це дослідження може надати цінну інформацію як розробникам, так і користувачам програмного забезпечення з відкритим кодом. Для розробників дослідження може запропонувати рекомендації щодо створення стійких бізнес-моделей, які підтримують постійний розвиток та інновації. Для користувачів дослідження може допомогти у прийнятті рішень про те, яке програмне забезпечення з відкритим кодом використовувати, виходячи зі сталості та довготривалості проєкту.

Загалом, це дослідження сприятиме кращому розумінню бізнес-моделей монетизації IT-проєктів з відкритим кодом і дасть уявлення про те, як ці моделі можна оптимізувати для підтримки постійного зростання прибутків та успіху програмного забезпечення з відкритим кодом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз особливостей розробки ІТ-проєктів відкритим кодом

Програмне забезпечення з відкритим вихідним кодом (ПЗВВК) відіграє все більшу роль в діяльності підприємств та урядів. Опитавши організації з 500 і більше працівниками або з доходами понад 500 мільйонів доларів, Фонд Linux виявив, що використання Linux для критично важливих робочих навантажень зросло з 60% у 2010 році до 73% у 2012 році [4]. Загалом, бізнес вказує на те, що покращення якості ПЗВВК протягом багатьох років зробило ПЗВВК кращим за пропрієтарні альтернативи у багатьох контекстах впровадження [5]. У державному секторі Міністерство оборони США виступило на підтримку ПЗВВК, формалізувавши свою позицію в плані "Розвиток відкритих технологій", який робить відкритість пріоритетом як для внутрішнього, так і для зовнішнього програмного забезпечення [6]. Уряд Великої Британії нещодавно також надав чітку перевагу відкритому програмному забезпеченню у своєму Посібнику з проєктування державних послуг [7]. Уряди все частіше усвідомлюють, що перехід на ПЗВВК дає змогу переорієнтувати "витрати" на ІТ з пропрієтарних продуктів на професійні послуги [8]. Якість ПЗВВК залежить від зусиль та інвестицій як розробників, так і тих, хто робить внесок у її розвиток. Зусилля розробників, спрямовані на покращення якості ПЗВВК, зазвичай розподіляються між різними, але необхідними видами діяльності. Відкрита природа ПЗВВК - це механізм, який дає можливість учасникам значно підвищити якість програмного забезпечення за допомогою оновлень, виправлень і нових функцій, розподіляючи витрати на розробку між усіма учасниками, які отримують безпосередню вигоду від існування програмного забезпечення [8]. Багато з цих стратегічних учасників використовують свої інвестиції та досвід, пропонуючи професійні послуги.

Насправді, надання послуг з доданою вартістю було основним джерелом доходу для комерційних компаній з ВПЗ. Сьогодні компанія Red Hat отримує

понад 1 мільярд доларів доходу за послуги підтримки на основі підписки, в основному завдяки двом своїм флагманським продуктам: Red Hat Enterprise Linux та JBoss [9]. Cloudera, яка надає дистрибутив Apache Hadoop з відкритим вихідним кодом під назвою CDH, інвестує значні кошти в проєкти з відкритим вихідним кодом, що входять до складу CDH, і так само покладається на ринок послуг для отримання прибутку. 1 Ці приклади підкреслюють, що високоякісне програмне забезпечення саме по собі часто не приносить користі організації, якщо воно недостатньо інтегроване з бізнес-процесами, щоб створювати цінність. Для того, щоб організації могли досягти такої високої доданої вартості, вони потребують послуг, і саме фірми, які мають великий досвід, мають найкращі можливості допомогти, як у випадку з Red Hat та Cloudera на їхніх відповідних ринках. Хоча є багато фірм, які також можуть надавати базову підтримку та послуги для Linux та Apache Hadoop, якість, пов'язана з отриманням послуг від таких фірм, як Red Hat та Cloudera, є значно вищою завдяки їх значним інвестиціям у розвиток цієї експертизи. Завдяки ринку послуг фірми мають економічно життєздатну бізнес-модель, яка може виправдати інвестиції в розробку ПЗВВК, навіть якщо саме програмне забезпечення, по суті, доступне безкоштовно.

Для розробника програмного забезпечення важливою перевагою використання відкритого вихідного коду є те, що якість його продукту з відкритим вихідним кодом використовує інвестиції спільноти, що, в свою чергу, збільшує цінність, яку отримують споживачі за контрактами на надання послуг. Однак, деякі учасники ПЗВВК також можуть бути вмотивовані інвестувати в розробку, щоб здобути досвід і конкурувати за ці контракти на обслуговування. Таким чином, розробник програмного забезпечення, який переходить на відкритий вихідний код, стикається з компромісом між підвищенням якості програмного забезпечення і послуг завдяки внескам спільноти і зростанням конкуренції на ринку послуг з боку цих зовнішньо вмотивованих учасників, а також численних постачальників базових пакетів послуг. З іншого боку, розробник програмного забезпечення може просто

дотримуватися більш традиційного, пропрієтарного підходу, коли він не отримує прямої вигоди від зусиль сторонніх розробників через закритість вихідного коду, але може отримувати прибуток, продаючи копії програмного забезпечення.

Однак, є кілька переваг, пов'язаних з тим, що розробники програмного забезпечення обирають шлях з відкритим кодом. По-перше, ширша участь у розробці програмного забезпечення, яка характеризує ПЗ з відкритим кодом, може значно підвищити якість, водночас утримуючи витрати на розробку на відносно низькому рівні, оскільки розподілені витрати розподіляються більш ефективно [10]. По-друге, коли активні учасники відкритого програмного забезпечення інвестують у розробку та експертизу, зростає конкуренція на ринку послуг, що також може бути корисним для споживачів.

По-третє, на відміну від пропрієтарного програмного забезпечення, розробник відкритого програмного забезпечення не має (практично) можливості стягувати плату за саме програмне забезпечення, оскільки воно є відкритим і його можна вільно завантажити. Ця додаткова можливість ціноутворення, яку зберігає розробник пропрієтарного програмного забезпечення, негативно впливає як на споживчий надлишок, так і на стимули потенційних постачальників послуг. Загалом, якщо заохочувати розвиток програмного забезпечення з відкритим кодом на певних ринках програмного забезпечення, то це може призвести до значного підвищення суспільного добробуту.

З іншого боку, через посилення конкуренції на ринку послуг у автора може зменшитися стимул інвестувати в розробку, оскільки рентабельність його інвестицій може бути нижчою. Зменшення інвестицій може призвести до зниження якості рішень, що виводяться на ринок, якщо це не буде адекватно компенсовано збільшенням внесків з боку спільноти. Тому, незважаючи на багатообіцяючі переваги, пов'язані з ПЗВВК, дуже важливо розуміти, як стратегічна взаємодія між розробником та учасниками ВПЗ в кінцевому підсумку визначає ринкові результати.

1.2 Постановка задачі

Об’єктом дослідження є монетизація ІТ-проєктів з відкритим кодом.

Предметом дослідження є бізнес-моделі монетизації ІТ-проєктів з відкритим кодом.

Метою даної роботи є дослідження методів та особливостей монетизації ІТ-проєктів з відкритим кодом. які використовуються у прогнозуванні прибутковості ІТ-проєкту з відкритим кодом, їх реалізація та використання у різних етапах планування та управлінням розробкою ІТ-проєкту.

У роботі необхідно дослідити як вартість розробки, атрибути ліцензування та тип проєкту впливають на рішення розробника програмного забезпечення щодо вибору стратегії розробки з відкритим кодом. Для розробки проблеми вибору розробника моделі мають враховувати економічні стимули розробників відкритого коду, які також конкурують на ринку послуг і чиї стратегічні інвестиції можна розглядати як взаємодоповнюючі та конкурентні. Використовуючи створені модель, необхідно вивчити вплив та стимули як розробника, так і подальшого учасника інвестувати в розвиток ВПЗ.

Для реалізації цієї мети в роботі необхідно:

— дослідити особливості розробки ІТ-проєктів з відкритим кодом;

— дослідити існуючі бізнес процеси монетизації ІТ-проєктів з відкритим кодом;

— обрати методи монетизації ІТ-проєктів та описати їх застосовуючи сучасні методи математичного моделювання;

— виконати практичне застосування розроблених моделей.

АНАЛІЗ МЕТОДІВ МОНЕТИЗАЦІЇ ІТ-ПРОЄКТІВ З ВІДКРИТИМ КОДОМ

2.1 Аналіз методу подвійного ліцензування

Під ліцензуванням ПЗВВК мається на увазі умови, які регулюють продукт ПЗВВК, що зазвичай визначають правила, за якими кінцевий користувач або розробник програмного забезпечення може використовувати, модифікувати та/або розповсюджувати програмне забезпечення. Ліцензії OSS суттєво відрізняються за рівнем обмежень. Найпоширенішими широкими формами ліцензування є GNU GPL (General Public License) та BSD (Berkeley Software Distribution), причому GPL є, мабуть, найпопулярнішою ліцензією [11]. GPL базується на понятті "копілефт", яке вимагає, щоб похідні роботи програмного забезпечення також приймали ліцензію GPL. Згідно з цією ліцензією, весь код, розроблений іншими на основі програмного забезпечення розробника, має бути загальнодоступним, в тому числі для використання розробником. Таким чином, ліцензія на кшталт GPL є відносно більш обмежувальною ліцензією і дає автору більшу пряму вигоду від зусиль учасників. Прикладом цього є обов'язкове прийняття GPL компанією Red Hat [12].

З іншого боку, ліцензії на кшталт BSD є дуже дозвільними, накладаючи мінімальні обмеження на використання та розповсюдження програмного забезпечення [13]. Згідно з такими ліцензіями, учасники не зобов'язані робити свій код, розроблений на основі коду розробника, доступним для громадськості, натомість дозволяючи учасникам зберігати права на свої модифікації та вдосконалення. Таким чином, ліцензії на кшталт BSD є менш обмежувальними, а вигода розробника від внеску інших обмежена порівняно з вигодою під GPL. PostgreSQL і Фонд програмного забезпечення Apache є двома відомими прихильниками цього стилю ліцензій [14]. Між цими двома крайнощами лежить низка інших ліцензій, таких як GNU LGPL (Lesser GPL),

яка послаблює обмеження на програми, що посилаються лише на двійкові файли коду під ліцензією GPL [13].

Різні ліцензії мають свої переваги та недоліки. Наприклад, Стюарт та ін.[15] досліджують вплив вибору ліцензії ПЗВВК на зацікавленість користувачів, розглядаючи сприйняття користувачами витрат і переваг використання програмного забезпечення, а також ризики, пов'язані з правовою невизначеністю та можливістю примусового виконання. Лернер і Тіроль [16] вказують на "викрадення" з боку розробників/постачальників комерційного програмного забезпечення, тобто зовнішніх учасників, які використовують відкритий програмний код для розробки (і потенційно підриву) комерційного програмного забезпечення з метою отримання прибутку під необмеженими ліцензіями.

Більш обмежувальні ліцензії надають перевагу розробникам ВПЗ, які можуть отримати вигоду від розробок наступних учасників. З іншого боку, менш обмежувальні ліцензії можуть забезпечити сильніші стимули для цих учасників до створення таких розробок [17]. Однак, через стратегічну природу внесків, те, як ліцензування впливає на інвестиції в рівновагу, може бути важко з'ясувати. Враховуючи, що комерційні фірми все більше інвестують у продукти ВПЗ та роблять внески у їх розробку, наша мета - дослідити та зосередитися на питаннях, пов'язаних з економічними стимулами для участі у проєктах ВПЗ. Оскільки вивчається програмне забезпечення з відкритим вихідним кодом з точки зору ринку послуг, рівень витрат на обслуговування є важливим для аналізу. Під витратами на обслуговування маються на увазі змінні витрати, які несуть фірми, надаючи послуги користувачам програмного забезпечення. Ці витрати можуть значно відрізнятися залежно від класу програмного забезпечення, про яке йдеться. Наприклад, інтеграція системи планування ресурсів підприємства (ERP) або системи управління взаємовідносинами з клієнтами (CRM) вимагатиме високих витрат на обслуговування, оскільки це, як правило, трудомісткий проєкт, що передбачає значну кастомізацію [18]. З іншого боку, підтримка реалізації MySQL як

частини стеку серверів додатків вимагатиме менших витрат на обслуговування, а навчання та підтримка інструментів підвищення продуктивності, таких як OpenOffice, обійдеться провайдеру ще дешевше.

Подвійне ліцензування - популярна бізнес-модель у світі вільного/відкритого програмного забезпечення. Вона передбачає розповсюдження одного і того ж програмного коду під двома різними ліцензіями: Загальною публічною ліцензією та комерційною ліцензією. Ця модель зазвичай використовується компаніями, які виробляють інструменти та програмне забезпечення, орієнтовані на розробників [19].

Успіх подвійного ліцензування можна пояснити положенням GPL про сильний зв'язок, яке вимагає, щоб будь-які похідні роботи або програмне забезпечення, які безпосередньо пов'язані з оригінальним програмним забезпеченням, також ліцензувалися за GPL. Однак компанії, які не бажають випускати своє програмне забезпечення під GPL, можуть придбати комерційну ліцензію, яка передбачає виняток з положення про зв'язування. Для прихильників ідеї OSS про "свободу слова" подвійне ліцензування розглядається як хороший компроміс. Воно дозволяє тим, хто дотримується GPL, отримати доступ до програмного забезпечення безкоштовно і зробити своє програмне забезпечення доступним як ПЗВВК, в той же час надаючи можливість тим, хто бажає зберегти код пропрієтарним, зробити це за допомогою комерційної ліцензії.

Однак модель подвійного ліцензування має і свої недоліки. Зовнішні розробники програмного забезпечення повинні також прийняти той самий режим ліцензування, що може зменшити обсяг зовнішніх внесків. Це може обмежити розробку програмного забезпечення виправленням помилок і невеликими доповненнями, оскільки зовнішні учасники можуть не бажати вносити значні зміни в рамках системи подвійного ліцензування.

Незважаючи на це, подвійне ліцензування залишається популярною бізнес-моделлю для компаній, які виробляють орієнтовані на розробників інструменти та програмне забезпечення. Вона пропонує компроміс між

перевагами ПЗВВК і можливістю підтримувати пропрієтарний код, дозволяючи компаніям обслуговувати як спільноти з відкритим кодом, так і пропрієтарні ринки.

2.2 Аналіз методу розробки комерційних розширень

Бізнес-модель розділених OSS/комерційних продуктів - це стратегія, коли компанія пропонує базову версію свого продукту на основі відкритого програмного забезпечення (OSS) поряд з комерційною версією, яка включає в себе пропрієтарні плагіни. Версія ПЗВВК зазвичай поширюється під дозвільною ліцензією, такою як Mozilla Public License, яка дозволяє змішувати її з пропрієтарним кодом [19]. Ця модель особливо приваблива для компаній, які хочуть скористатися перевагами розробки ПЗВВК, зберігаючи при цьому певний контроль над комерціалізацією програмного забезпечення.

Однією з головних переваг цієї моделі є те, що вона дозволяє ширше залучати зовнішніх учасників, оскільки не потрібно погоджуватися на подвійне ліцензування. Крім того, версія ПЗВВК може слугувати полігоном для тестування нових функцій та ідей, які згодом можуть бути інтегровані в комерційну версію. Однак, ця модель має деякі невід'ємні недоліки, які необхідно враховувати.

По-перше, продукту ПЗВВК має бути цінним, щоб привабити користувачів, але він не повинен бути достатньо повним, щоб запобігти конкуренції з комерційною версією. Такого балансу може бути складно досягти і підтримувати протягом тривалого часу. Крім того, якщо програмне забезпечення викликає значний інтерес, розробники можуть спробувати доповнити відсутню функціональність виключно за допомогою відкритого коду, що знизить привабливість комерційної версії.

Компанії, які приймають цю модель, також повинні бути обережними, щоб уникнути фрагментації своєї користувацької бази між ПЗВВК і комерційною версіями. Важливо переконатися, що відмінності між двома версіями є чіткими і що користувачі розуміють, що вони отримують з кожною версією.

Інша проблема полягає в тому, що ця модель може не підходити для всіх типів програмного забезпечення. Наприклад, вона може добре працювати для інструменту розробки програмного забезпечення або системи управління контентом, але може бути не найкращим вибором для складного корпоративного додатку.

Незважаючи на ці виклики, модель розділеної OSS/комерційних продуктів виявилася успішною для багатьох компаній. Одним з найяскравіших прикладів є база даних MySQL, яка пропонує громадську версію під GPL та комерційну версію з власними плагінами. Інші приклади включають Red Hat, яка пропонує безкоштовну версію свого дистрибутиву Linux поряд з комерційною версією з підтримкою і додатковими функціями, і SugarCRM, яка пропонує версію для спільноти під GPL і комерційну версію з додатковою функціональністю.

Модель розділених відкритих комерційних продуктів може бути ефективним способом для компаній використовувати переваги розробки OSS, зберігаючи при цьому певний контроль над комерціалізацією. Однак, необхідно ретельно проаналізувати програмне забезпечення, що розробляється, базу користувачів і баланс між відкритими і комерційними версіями.

2.3 Аналіз моделі монетизації послуг та підтримки

Модель підтримки та обслуговування полягає в тому, що компанія або організація, що стоїть за проєктом з відкритим кодом, надає підтримку та послуги клієнтам, які користуються програмним забезпеченням. Це опирається на принцип, що хоча програмне забезпечення є безкоштовним, користувачам все одно може знадобитися допомога у встановленні, налаштуванні або обслуговуванні. Компанії можуть пропонувати пакети підтримки, які включають технічну допомогу, усунення недоліків та навчання, серед інших послуг.

Однією з головних переваг моделі підтримки та послуг є те, що вона забезпечує стабільний потік доходів для компанії чи організації, що стоїть за проєктом з відкритим кодом. Поки користувачі продовжують використовувати програмне забезпечення, вони, ймовірно, потребуватимуть певного рівня підтримки та послуг, які можна монетизувати. Крім того, ця модель може допомогти побудувати довіру з клієнтами, оскільки вони знають, що можуть покладатися на компанію чи організацію в питаннях постійної підтримки.

Ще однією перевагою моделі підтримки та послуг є те, що вона може сприяти інноваціям і розробці нових функцій для програмного забезпечення з відкритим вихідним кодом. Пропонуючи доповнення та плагіни для продажу, компанії можуть інвестувати в розробку нових функцій і можливостей, які в іншому випадку були б неможливими. Це може допомогти залучити нових користувачів та утримати існуючих користувачів у програмі.

Однак у моделі підтримки та обслуговування є й певні потенційні недоліки. Однією з головних проблем є визначення правильної ціни на підтримку та послуги. Якщо ціна занадто висока, клієнти можуть вирішити звернутися за підтримкою до іншої компанії або перейти на інше програмне рішення. З іншого боку, якщо ціна занадто низька, компанія чи організація

може не отримати достатнього доходу, щоб підтримувати розробку та підтримку програмного забезпечення з відкритим вихідним кодом.

Крім того, модель підтримки та послуг може не підходити для всіх типів програмного забезпечення з відкритим кодом. Деякі програмні проєкти можуть не потребувати постійної підтримки або не мати достатньо великої бази користувачів, щоб підтримувати монетизацію підтримки та послуг. У цих випадках альтернативні бізнес-моделі, такі як пожертви, спонсорство або ліцензування, можуть бути більш доречними.

2.4 Аналіз моделі розробки платформ

Розробка платформ з точки зору компанії полягає в надання вибору програмних проєктів, а також підтримку, інтеграцію та послуги для створення протестованої та перевіреної платформи. У цьому сенсі навіть дистрибутиви Linux можна вважати платформами. Цікавим спостереженням є те, що багато з цих дистрибутивів ліцензуються за ліцензіями, що максимізувати зовнішній внесок і використовувати захист авторських прав для запобігання прямому копіюванню, але не клонуванню, яке передбачає видалення захищених авторським правом матеріалів, таких як логотипи і торгові марки, для створення нового продукту [20].

Основна перевага постачальників платформ полягає в тому, що вони пропонують гарантовану якість, стабільність і надійність, а також впевненість у підтримці критично важливих для бізнесу додатків. Надаючи випробувану і перевірену платформу, постачальники платформ можуть зменшити ризик і складність, пов'язані з вибором та інтеграцією різних програмних компонентів. Це може бути особливо цінним для підприємств, які потребують розгортання великомасштабних систем і потребують високого ступеня довіри до програмного забезпечення, яке вони використовують.

Постачальники платформ також можуть отримати вигоду від мережевих ефектів, коли цінність платформи зростає, від розробників і користувачів роблять свій внесок. Створюючи екосистему розробників, користувачів і партнерів, постачальники платформ можуть стимулювати інновації та прискорювати розробку нових функцій і можливостей. Це, в свою чергу, може залучити більше користувачів і клієнтів, створюючи цикл зростання.

2.5 Порівняння розглянутих моделей монетизації

Таблиця 2.1 – Порівняльний аналіз різних моделей монетизації

Модель	Переваги	Недоліки
Подвійне ліцензування	– дозволяє розробникам використовувати два різних типи ліцензій: відкриту ліцензію для спільноти відкритого ПЗ і пропрієтарну ліцензію для комерційних клієнтів; – забезпечує розробникам можливість отримувати прибуток від комерційних клієнтів, які не хочуть розповсюджувати свої розширення або модифікації з використанням відкритої ліцензії.	– складніше управління ліцензіями, оскільки потрібно враховувати два різних типи ліцензій та їх обмеження; – можливість конфліктів між відкритою спільнотою та комерційними клієнтами
Розробки комерційних розширень	– розробка додаткових комерційних розширень, що надають додаткові можливості; – можливість отримання прибутку від продажу цих комерційних розширень або підписки на них.	– залежність від постійної розробки нових розширень, щоб зберегти прибуток; – можливість конкуренції зі сторонніми розробниками, які також можуть створювати подібні комерційні розширення.

Кінець таблиці 2.1

Модель	Переваги	Недоліки
Монетизація послуг та підтримки	– пропозиція послуг з інсталяції, налаштування та підтримки для відкритого ПЗ, що збільшить задоволеність клієнтів; – можливість отримання прибутку від надання послуг та підтримки користувачам.	– залежність від постійного залучення нових користувачів та утримання існуючих для забезпечення стабільного потоку доходу; – конкуренція з іншими компаніями, які також можуть надавати аналогічні послуги.
Розробка відкритих платформ	– побудова платформи або екосистеми, яка забезпечує спільноту розробників та сторонніх розробників можливість розробляти та продавати свої додатки або розширення; – заробіток на комісіях від продажу у магазині або маркетплейсі платформи.	– велика конкуренція на платформі від інших розробників, які також пропонують свої додатки або розширення; – потреба в постійному підтриманні та оновленні платформи для забезпечення якості та безпеки.

В роботі буде проведено детальне дослідження моделі монетизації послуг та моделі розробки відкритих платформ, оскільки вони дозволяють різні джерела надходжень прибутків, а також дослідити як між фірмову конкуренцію, так і конкуренцію в рамках одного ІТ-проєкту між автором та контрибуторами відкритого ПЗ.

3 ДОСЛІДЖЕННЯ МОДЕЛІ ОТРИМАННЯ ПРИБУТКУ ВІД КОНСУЛЬТАЦІЙНИХ ПОСЛУГ

У цій частині роботи представляється модель, яка відображає економічні стимули для фірм робити внесок у розробку програмного забезпечення з відкритим кодом. Досліджується спільна розробка та цінова рівновага на ринках програмного забезпечення для двох стратегій - відкритого та пропрієтарного програмного забезпечення. Якщо фірма обирає стратегію з відкритим кодом, вона робить свій вихідний код загальнодоступним, що має кілька наслідків.

По-перше, програмне забезпечення отримує поштовх до розвитку, оскільки зовнішні учасники допомагають писати та вдосконалювати вихідний код продукту, а також створювати супутні модулі та утиліти.

По-друге, фірма має значною мірою покладатися на послуги як на основне джерело доходу, оскільки її вихідний код є у вільному доступі для користувачів. По-третє, наявність вихідного коду полегшує конкурентам на ринку послуг накопичення досвіду та покращення якості власних пропозицій, а це означає, що фірма-розробник стикається з жорсткішою конкуренцією на цьому ринку. Наприклад, такий великий учасник, як IBM, може присвятити величезні ресурси розробці проєкту з відкритим вихідним кодом, щоб підвищити рівень володіння програмним забезпеченням і вдосконалити свої пакети послуг. Ці інвестиції можуть як принести користь, так і зашкодити розробнику, який працює у середовищі з відкритим кодом. Крім того, фірма-розробник може обрати варіант пропрієтарного коду, і в цьому випадку вихідний код її продукту залишається закритим.

У цьому випадку фірма несе всі витрати на розробку продукту самостійно. З іншого боку, на додаток до потенційного стягнення плати за програмне забезпечення та інтеграційні послуги, фірма зберігає можливість стягувати ціну за кожну копію продукту, незалежно від того, чи надає вона

послуги, чи ні. Використовуючи цю модель, буде досліджено два основні питання: По-перше, визначена рівновагу на ринку програмного забезпечення, де розробники та користувачі інвестують у розробку, а фірми конкурують за доходи від надання послуг, використовуючи стратегії ліцензування з відкритим кодом та пропрієтарне ліцензування.

По-друге, спираючись на цей аналіз рівноваги, досліджується, як ефективність розробки програмного забезпечення як для розробника, так і для наступних учасників впливає на вибір розробником стратегії ліцензування вихідного коду.

По суті, в основі роботи лежить управління двома взаємодоповнюючими товарами, а саме: програмним забезпеченням та інтеграцією і послугами, що надаються для нього. Основний компроміс у цьому рішенні щодо ліцензування програмного забезпечення з послугами полягає в наступному: розробник може або розробити продукт самостійно і натомість мати можливість брати плату за програмне забезпечення (пропрієтарна стратегія), або він може відкрити вихідний код для зовнішніх учасників, щоб розділити тягар інвестицій у розробку, але покладатися лише на доходи від послуг та інтеграції, хоча, можливо, з вищою якістю програмного забезпечення та послуг завдяки зусиллям фірми, що надає послуги (стратегія з відкритим вихідним кодом).

Існує низка факторів, які відіграють вирішальну роль у визначенні найкращої стратегії ліцензування, таких як відносна ефективність фірми-розробника та фірми-контрибутора, а також можливості фірм, які отримують вигоду від програмних розробок одна одної. Ефективність контрибутора - це палиця з двома кінцями для розробника. Коли учасник має високу ефективність розробки і використовує розробки для покращення якості, це може окупитися для розробника відкриттям вихідного коду, оскільки це може призвести до значного підвищення загальної якості програмного забезпечення та послуг при невеликих витратах для розробника. Однак, потужний учасник може також бути грізним конкурентом, який може вкрасти значну частку

ринку. Приймаючи рішення про ліцензування, розробник повинен збалансувати ці два фактори. Було виявлено, що якщо розробник достатньо ефективний у розробці, його рішення щодо ліцензування залежить головним чином від його здатності використати розробку учасника для покращення якості: якщо розробник вміє покращувати якість свого програмного забезпечення/пакету послуг, використовуючи розробку учасника, то оптимальною є стратегія відкритого коду, якщо ж ні, то розробнику краще залишити програмне забезпечення пропрієтарним. Виялено, що збільшення витрат розробника на розробку може фактично зменшити ціну на послуги розробника і збільшити ціну на послуги учасника. Це відбувається тому, що зниження ефективності розробки автора зменшує інвестиційні зусилля автора та збільшує диференціацію між пропозиціями автора та контрибутора. Отже, постачальник може виграти, збільшивши свої зусилля в розробці, підвищивши якість та ціну.

3.1 Опис моделі монетизації ІТ-проєктів шляхом продажу продукту та послуг

Розробник програмного забезпечення визначає, чи слідувати стратегії ліцензування пропрієтарного або відкритого програмного забезпечення, яку позначено через $\rho \in \{\mathrm{P}, \mathrm{O}\}$. Індекс P використовується для позначення стратегії ліцензування пропрієтарного програмного забезпечення та O для позначення стратегії ліцензування відкритого ПЗ. Після прийняття рішення про ліцензування розробник інвестує в проєктування та розробку програмного продукту. Вторинна фірма, контрибутор, може вирішити докласти зусиль докласти зусиль, щоб інвестувати в розробку і підвищити якість програмного рішення. Першопочаткового автора розробника позначено індексом o, а контрибутора - індексом c. У моделі зосереджено увагу на розробнику та

контрибуторі, які мають намір отримувати прибутки від надання послуг з інтеграції програмного забезпечення та інших послуг споживачам. Важливо також зазначити, що зазвичай є й інші учасники розробки відкритого програмного забезпечення, які не мають комерційних мотивів. Ці учасники мають нестратегічні мотиви, такі як хобі, альтруїзм та ідеалізм. Ці нестратегічні внески можна врахувати нестратегічні внески спільноти відкритого програмного забезпечення як екзогенну, випадкову добавку компонент якості програмного забезпечення та послуг для всіх постачальників, але ця проста арифметична дія не змінить природу висновків або результатів моделі. Для того, щоб зробити модель простішою, цей компонент не включений в модель та аналіз.

Після придбання або прийняття рішення про використання програмного забезпечення споживач повинен встановити та інтегрувати його, тобто отримати послугу, щоб воно стало придатним для використання, а споживач міг отримати від нього користь. Цінність, яку споживач отримує від використання програмного забезпечення, залежить від багатьох факторів, зокрема, від характеристик продукту, надійності, інтеграції з бізнес-системами та процесами, а також від підтримки. У свою чергу, якість (а отже, і цінність для споживача) таких функцій, як інтеграція та підтримка, залежить від постачальника послуг. Фірма-постачальник послуг з більшим досвідом, кращими інструментами підтримки та утилітами, а також кращою підготовкою забезпечує вищу якість послуг. Кожна фірма, яка надає послуги з інтеграції та обслуговування, по суті, пропонує свій власний рівень якості послуг, і загальна цінність, яку споживач отримує від використання програмного забезпечення, залежить від фірми, з якою він укладає контракт на інтеграцію та обслуговування [20]. Фірма може покращити свій досвід та загальну якість, інвестуючи зусилля в ці сфери. Тому, хоча за обома стратегіями ліцензування обидві фірми використовують один і той самий програмний продукт, через потенційну різницю в якості послуг, загальна якість пропозиції фірми, яку визначено як сукупну якість використовуваного

програмного забезпечення та послуг, що надаються цією фірмою, може відрізнятися між цими двома фірмами.

Загальна якість для автора позначена через Q_o та загальна якість для котрибутора через Q_c. Загальна якість пропозиції фірми, що робить внесок, певною мірою залежить від зусиль, вкладених розробником програмного забезпечення у розробку продукту. Крім того, якщо продукт має відкритий вихідний код, додаткові розробки контрибутора є доступними для громадськості, тому в цьому випадку загальна якість пропозиції розробника також виграє від зусиль фірми, що зробила внесок. Таким чином, ринок програмного забезпечення та супутніх послуг з можливістю ліцензування з відкритим кодом має цікаву особливість: розробник і учасник доповнюють один одного в поліпшенні загальної якості програмного забезпечення та послуг, в той час як вони також конкурують між собою на ринку за надання цих послуг. Політика щодо програмного забезпечення, обрана розробником (тобто, пропрієтарне чи відкрите програмне забезпечення) змінює стимули для цих двох фірм, а отже, впливає на їхній стратегічний вибір зусиль впливає на їхній стратегічний вибір щодо зусиль, вкладених у програмне забезпечення.

Існує множина споживачів, які мають неоднорідні вподобання щодо загальної якості комплексного програмного рішення, яке вони отримують. Ця характеристика типу споживача моделюється як рівномірно розподілена, $\theta \in \Theta = [0,1]$, де θ позначає чутливість конкретного споживача до якості або "тип". Споживач з типом θ отримує цінність θQ_o, якщо він укладає контракт з автором, і θQ_c, якщо він укладає контракт з контрибутором.

Розробник обирає зусилля в розробці e_o несе квадратичну опуклу вартість зусиль $\frac{\beta_O e_O^2}{2}$. Аналогічно, дописувач інвестує зусилля e_C за ціною $\frac{\beta_C e_C^2}{2}$.після інвестиційного рішення автора. Зусилля у розробці відкритого коду складаються з праці та ресурсів, і, особливо в індустрії програмного забезпечення, фірми демонструють значні відмінності в розмірах та здібностях працівників, а також відмінності в тіньових цінах на час працівників та

доступність ресурсів для будь-якого проєкту. Таким чином, β_O і β_C загалом відрізняються, і фірми демонструють неоднорідність у цьому вимірі [21]. Також для автора e_oозначає додаткові зусилля в розробці програмного забезпечення понад базову суму, необхідну для випуску програмного забезпечення в мінімально прийнятній формі.

Залежно від рішення щодо політики програмного забезпечення (O чи P) на пропозицію фірми позитивно впливатимуть інвестиції інших фірм у загальнодоступні відкриті вихідні коди, компоненти сервісної підтримки, внески в утиліти та інформацію. За політики відкритого коду (тобто, для $\rho = O$), загальна якість, яку пропонує розробник, залежить не лише від його власних зусиль e_o^O, але й від зусиль, докладених учасниками, так що

$$Q_o^O = s_{oo}^O e_o^O + s_{oc}^O e_c^O , \tag{3.1}$$

де s_{oo}^O– коефіцієнт впливу зусиль автора на його загальну якість;

s_{oc}^O – коефіцієнт міжфірмового впливу зусиль контрибутора на загальну якість автора;

e_o^O– рівень зусиль автора щодо розробки ПЗ;

e_c^O –рівень зусиль контрибутора щодо розробки ПЗ.

Аналогічно, загальна якість для контрибутора залежить від зусиль автора і задається формулою

$$Q_c^O = s_{co}^O e_o^O + s_{cc}^O e_c^O , \tag{3.2}$$

де s_{co}^O– коефіцієнт впливу зусиль автора на загальну якість контрибутора;

s_{cc}^O – коефіцієнт впливу зусиль контрибутора на його загальну якість;

e_o^O– рівень зусиль автора щодо розробки ПЗ;

e_c^O –рівень зусиль контрибутора щодо розробки ПЗ

Параметри $s_{ij}^{O} > 0, i, j \in \{o, c\}$, вказують на вплив зусиль фірми j на загальну якість фірми i. Величини коефіцієнтів s_{ij}^{O} значною мірою визначаються конкретним ринком програмного продукту та характером фірм-розробників і фірм-донорів. Ці коефіцієнти мають суттєве економічне та стратегічне значення. Їх відносні величини є критично важливими для визначення стимулів фірм докладати зусиль до створення програмного забезпечення, оскільки конкурент фірми, який отримує вигоду від її зусиль за високими ставками, може обмежити її стимули нести витрати, пов'язані з такими зусиллями. Крім того, відносні величини цих коефіцієнтів між політикою пропрієтарного та відкритого програмного забезпечення можуть також впливати на ліцензійну політику розробника. Ці коефіцієнти відіграватимуть важливу роль у визначенні підсумків і результатів.

За пропрієтарної стратегії споживачі, які купують продукт, можуть вибирати, чи користуватися послугами розробника, чи послугами учасника. Оскільки за цієї стратегії вихідний код є закритим, зусилля учасника не приносять прибутку в компонент якості автора, тобто $s_{oc}^{P} = 0$ Примітно, що s_{oc}^{P}ос пов'язана з вигодою, яку автор може отримати від зусиль розробника. За пропрієтарної стратегії розробнику важче залучити зусилля розробників на свою користь. Наприклад, Microsoft може спробувати впровадити програми сертифікатів для підтримки нового програмного забезпечення. Деякі учасники можуть інвестувати свої зусилля, щоб отримати сертифікати. Однак це не має прямого відношення до покращення якості програмного забезпечення або до покращення якості послуг Microsoft. Отже, s_{oc}^{P}близьке до 0. Крім того, закритість вихідного коду обмежує можливості дописувача покращувати власні фактори якості та пов'язані з ними пропозиції, тобто $s_{cc}^{P} < s_{cc}^{O}$. Припускається, що $s_{oo}^{P} = s_{oo}^{O}$ є постійним і не залежить від стратегії фірми-розробника. Для того, щоб зробити справедливе порівняння між прибутковістю двох бізнес-моделей, вважається що цей параметр постійним і

виокремлюємо основні ефекти, які допомагають кожній моделі працювати відносно краще.

Для простоти аналізу припускається, що автор здатен краще використовувати свої власні зусилля в розробці для покращення якості, ніж учасник, тобто $s_{oo}^{O} > s_{co}^{O}$ та $s_{oo}^{p} > s_{co}^{P}$. У багатьох випадках розумно очікувати, що розробник програмного забезпечення може це зробити, тому що як творець програмного забезпечення, він, ймовірно, має фундаментальні знання та досвід, а також контроль над напрямком проєкту, що дозволяє йому краще використовувати власні зусилля.

Рішення про використання приймається споживачами на останньому етапі. За пропрієтарної стратегії автор встановлює ціну на продукт p^P і ціну на свої послуги p_o^P , тоді як контрибутор встановлює лише ціну на свої послуги, пропонуючи p_c^P . За стратегії з відкритим кодом, ціноутворення на інтеграцію та послуги все ще відбувається (p_o^P та p_c^P), але ціна продукту дорівнює нулю (p_o^P=0). Для спрощення в цій роботі припускається, що вартість надання послуг для фірм дорівнює нулю.

У пропрієтарному випадку споживач з типом θ обирає одну з наступних дій: не використовувати програмний продукт, придбати як продукт, так і послуги у розробника, або придбати продукт у розробника, отримуючи при цьому послуги від учасника. Його чистий прибуток від кожної з цих дій дорівнює

$$V^P(\theta) =$$

$$\begin{cases} Q_o^P\theta - p^P - p_o^p, & \text{якщо придбано продукт та послуги у автора;} \\ Q_c^P\theta - p^P - p_c^p, & \text{якщо придбано продукт та послуги у контибутора;} \\ 0, & \text{продукт не придбано} \end{cases} \quad (3.4)$$

Згідно зі стратегією відкритого коду, розробник відмовляється від доходів від продажу продуктів, зосереджуючись на ринку послуг. У цьому випадку чистий прибуток від споживача складається з

$$V^O(\theta) = \begin{cases} Q_o^O\theta - p_o^O, & \text{якщо придбано послуги у автора;} \\ Q_c^O\theta - p_c^O, & \text{якщо придбано послуги у контибутора;} \\ 0, & \text{продукт не придбано} \end{cases} \quad (3.5)$$

Загальна якість програмного забезпечення та послуг (Q_o^P, Q_o^P та Q_c^P, Q_c^O) відрізняються за кожною стратегією вихідного коду стратегії кодування через різні інвестиційні стимули, а також певні параметри перехресних ефектів якості s_{ij}^P та s_{ij}^O, які відрізняються для двох стратегій кодування. Підсумовуючи, часова шкала для моделі виглядає наступним чином:

— розробник обирає стратегію ліцензування, ρ ∈ {P, O}, яка робить продукт або пропрієтарним, або з відкритим з відкритим кодом;

— автор інвестує в розробку програмного забезпечення, e_o;

— знаючи, чи є програмне забезпечення пропрієтарним або з відкритим кодом, і спостерігаючи за зусиллями розробника у розробці ПЗ, учасник вирішує, чи пропонувати послуги для ПЗ та скільки інвестувати у його подальшу розробку. скільки інвестувати в подальшу розробку, e_c;

— розробник спостерігає за тим, чи є учасник на ринку, і якщо так, то який рівень розробки програмного забезпечення;

— обидві фірми одночасно встановлюють ціни на свої пропозиції, тобто розробник та учасник встановлюють ціни на послуги, p_o^p та p_c^p, відповідно, для ρ ∈ {O, P}. Крім того, якщо продукт є запатентованим, то виробник встановлює ціну за одиницю товару, p^p , також одночасно;

— кожен споживач обирає, чи використовувати продукт і у якої фірми купувати послуги.

3.2 Практичне застосування розробленої моделі

Як альтернативу традиційній пропрієтарній стратегії фірма-розробник програмного забезпечення може прийняти стратегію відкритого коду, за якої вихідний код є відкритим для інших суб'єктів, таких як розробники та користувачі. Однією з ключових особливостей визначення відкритого коду є те, що вихідний код програмного продукту повинен бути доступним [22].

Як бізнес-модель, відкритий підхід, який використовує знання та ресурси, що знаходяться у відкритій спільноті, має багато суттєвих переваг. Вартість розробки для розробника є нижчою, оскільки він може скористатися внеском інших розробників як для покращення програмного забезпечення, так і для написання утиліт та іншого допоміжного коду, корисного для інтеграції та супроводу продукту. При відкритому підході вихідний код надається у вільному доступі, і в цьому випадку стягування плати за сам базовий програмний продукт є неможливим. Однак, оскільки споживачі цінують послуги, фірми отримують більшу частину своїх доходів, надаючи послуги з інтеграції, підтримки та консультування.

3.2.1 Характеристика ринку споживачів відкритого програмного забезпечення

Маючи рівні витрат e_o^O та e_c^O , та цін p_o^O та p_c^O, кожен споживач типу $\theta \in \Theta$ обирає, з ким укласти контракт, щоб максимізувати свій чистий виграш, наведений у формулі 3.2, що відображає вибір, який він має для отримання послуг. Поведінка споживача в стані рівноваги знову є симетричною і залежить від того, хто пропонує вищу загальну якість виробник чи споживач. Доказ наведений в додатку А.

Рисунки 3.1-3.3 ілюструють споживчий ринок як це описано в додатку А коли автор є лідером за якістю, тобто $Q_o^O > Q_c^O$.

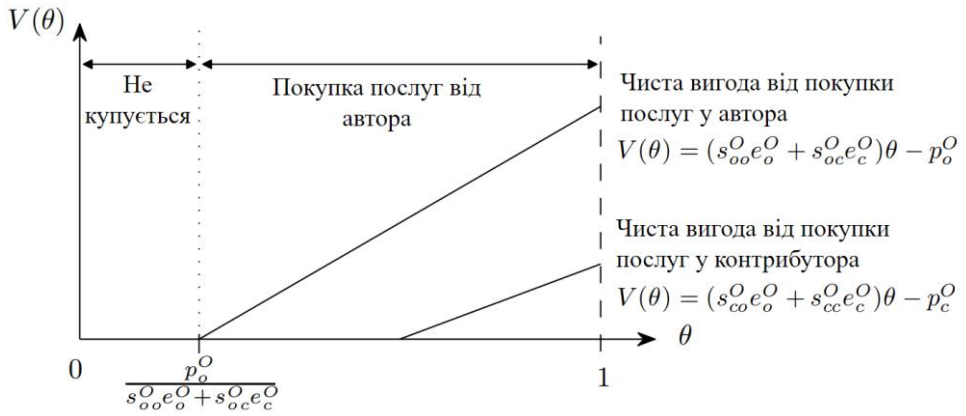

Рисунок 3.1 – Стан ринку, коли активний тільки автор продукту

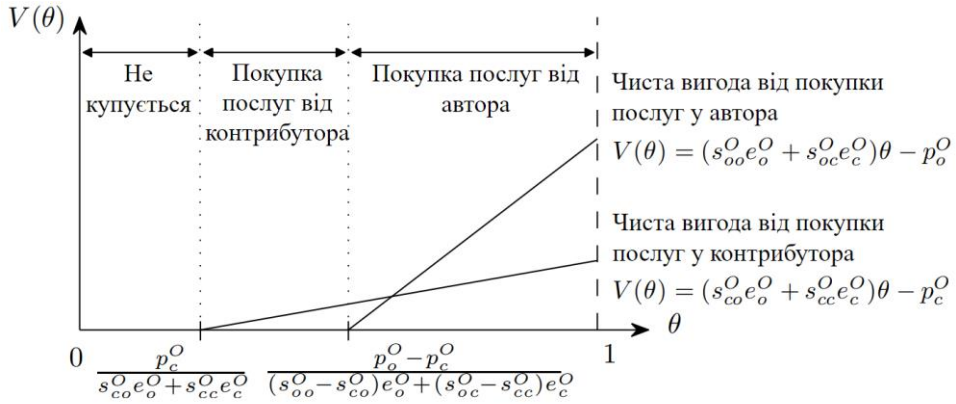

Рисунок 3.2 – Стан ринку, коли активний і автор і контрибутор продукту

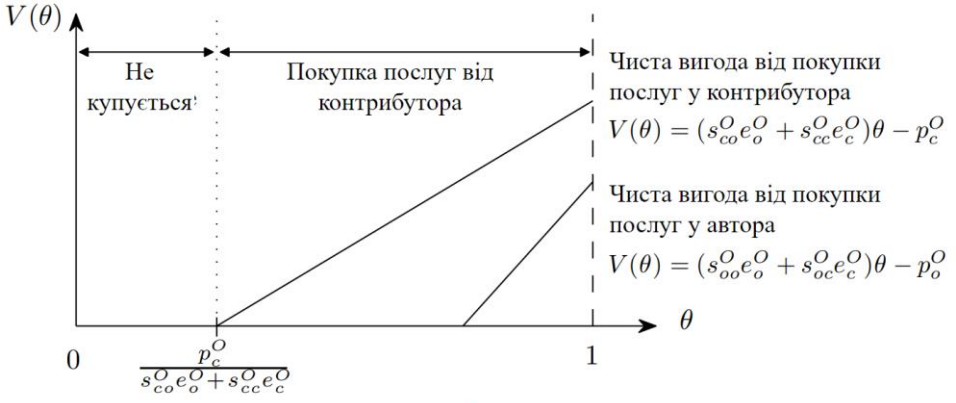

Рисунок 3.3 – Стан ринку, коли активний тільки контрибутор продукту

Якщо обидві фірми встановлять занадто високі ціни, жоден споживач не купуватиме послуги, а отже, жоден споживач не вирішить користуватися програмним забезпеченням. Якщо або учасник встановлює занадто високу ціну, або автор встановлює достатньо низьку ціну, щоб зробити свої послуги привабливішими, ніж у контрибутора, для всіх типів споживачів, то всі покупці укладуть контракт з автором, як це показано на рис.3.1. З іншого боку, якщо ціна жодної з фірм не є надто високою, а ціна оригінатора не є надто низькою, щоб витіснити постачальника з ринку, то обидві фірми ділять ринок послуг, причому постачальник обслуговує нижчий прошарок користувачів. Таку структуру споживчого ринку можна побачити на рис.3.2.

Нарешті, якщо ціна постачальника досить низька, а ціна розробника досить висока, то, незважаючи на те, що розробник має вищу якість, всі клієнти, які використовують програмне забезпечення, вважають за краще укладати контракт з постачальником, а розробник залишається поза ринком. Це можна побачити на рис.3.3. Результат для випадку, коли контрибутор є лідером за якістю, має точно таку ж структуру, за винятком того, що ролі розробника і контрибутора міняються місцями.

Враховуючи відповідні рівні зусиль, e_o^O та e_c^O , фірми конкурують на ринку послуг, встановлюючи відповідні ціни, щоб максимізувати свої прибутки, приймаючи ціни одна одної як задані. Результатом рівноваги Неша на останньому етапі гри є рівноважні ціни на послуги, які позначаються через $p_o^{O*}(e_o^O, e_c^O)$ та $p_c^{O*}(e_o^O, e_c^O)$, які є функціями від рівнів зусиль, обраних фірмами на попередніх етапах. Наступне твердження описує, як виникає споживча цінова рівновага та формується ринок. Доказ твердження наводиться в додатку Б.

Враховуючи рівні зусиль розробника та учасника, рівноважні результати ціноутворення на програмне забезпечення та послуги характеризують ринок наступним чином:

а) якщо або розробник, або учасник є лідером за якістю, то обидві фірми є активними на ринку послуг, кожна з яких обслуговує позитивну масу споживачів. Рівноважні ціни симетрично змінюються, коли роль лідера якості змінюється на протилежну;

б) якщо жодна з фірм не є лідером за якістю (тобто вони пропонують еквівалентну загальну якість), то фірми беруть участь у конкуренції Бертрана, яка призводить до того, що обидві фірми встановлюють ціни на рівні витрат у стані рівноваги.

Якщо якість послуг фірм не є однаковою, то в стані рівноваги виробник і споживач встановлюють ціни таким чином, щоб обидві фірми обслуговували позитивну частку ринку споживачів. Тобто, у стані рівноваги серед усіх можливих варіантів, показаних на рис.3.1, рис.3.2 та рис.3.3, може виникнути лише результат, показаний на рис.3.2, де обидві фірми беруть участь в активній дуополії з позитивними частками ринку. Для кожної фірми завжди існує певна вигода у встановленні невеликої позитивної ціни для того, щоб захопити частину споживачів з низьким рівнем доходу, а отже, жодна з фірм не може витіснити іншу з ринку. Як наслідок, у стані рівноваги фірми погоджуються на спільний олігополічний ринок.

З іншого боку, якщо якості фірм є рівними, то вони вступають у постійну конкуренцію, яка знижує їхні рівноважні ціни аж до собівартості надання послуг, що дорівнює нулю.

3.2.2 Визначення стратегії розповсюдження в залежності від вартості розробки

Одним з аргументів на користь підходу з відкритим вихідним кодом є те, що можна отримати значні потенційні вигоди, якщо використовувати зусилля зовнішніх розробників для створення якіснішого продукту. Ця

стратегія використовує внесок зовнішніх учасників для виявлення та виправлення помилок і вразливостей, а також для управління дизайном програмного забезпечення та його майбутнім напрямком.

Незважаючи на ці переваги, можливості постачальника програмного забезпечення для отримання прибутку стають обмеженими. Відкритість коду по суті означає, що програмний продукт є безкоштовним, тому будь-який споживчий надлишок, який виникає як прямий результат цінності, отриманої від самого базового продукту, важко привласнити. Отже, успіх бізнес-моделі з відкритим кодом має полягати в її здатності пов'язувати більшу частину внутрішньої цінності програмного забезпечення з компонентом послуг. Однак автор програмного забезпечення всеодно повинен конкурувати за привласнення цього надлишку на конкурентному ринку послуг. Доказ наведених тверджень в додатку В.

Коли витрати на розробку β_c контрибутора є достатньо низькими, стратегія відкритого коду є привабливою для розробника, тоді як коли витрати на розробку учасника є високими, привабливою є пропрієтарна стратегія. Коли ефективність витрат на розробку учасника є високою, тобто коли β_c є низьким, стратегія з відкритим кодом є привабливою для автора, оскільки він може використати його внесок, що значно підвищить якість його пропозиції завдяки перехресним ефектам.

Однак, оскільки витрати на розробку з боку контрибутора є низькими, він також є сильнішим конкурентом для автора на ринку послуг, який є його єдиним джерелом доходу при стратегії відкритого коду. Коли контрибутор є достатньо ефективним у розробці програмного забезпечення, вигоди автора від цього зовнішнього внеску домінують над втратами та зниженням прибутковості, пов'язаними з протистоянням сильному та ефективному конкуренту. З іншого боку, коли витрати учасника на розробку є високими, тобто коли β_c є високим, його зусилля з розробки будуть обмеженими, а вигоди, які автор може отримати від відкриття вихідного коду, є скромними порівняно з доходами, які він втратить від того, що не зможе стягувати плату

за саме програмне забезпечення. Тому, за низької ефективності розробки контрибуторами, оптимальною політикою для автора є збереження програмного забезпечення пропрієтарним.

Однак, підвищення ефективності контрибуторів не завжди означає стратегічний перехід до політики відкритого вихідного коду і навпаки. Насправді, підвищення ефективності може зробити політику пропрієтарності кращою для автора, як показано на рис.3.4.

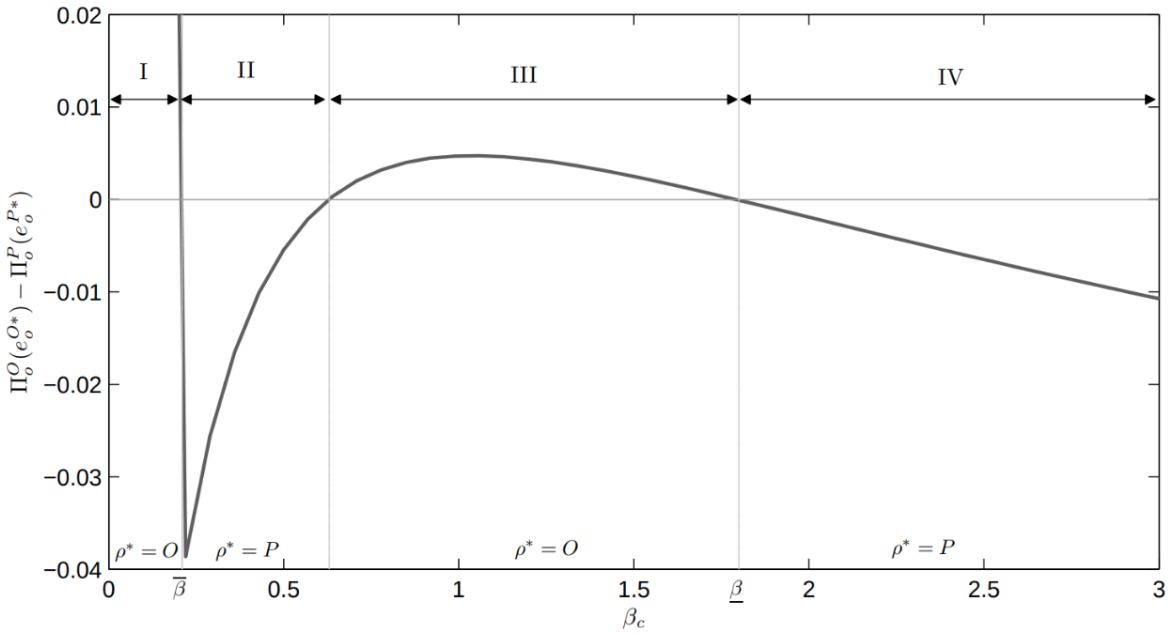

Рисунок 3.4 – Вплив витрат на розробку на вибір стратегій відкритого коду

На рис.3.4 показано різницю між рівноважним прибутком за стратегії відкритого коду та пропрієтарної стратегії $\Pi_o^O(e_o^{O*}) - \Pi_o^P(e_o^{P*})$ як функцію параметра ефективності витрат на розробку (β_c). Таким чином, на рис.3.4 позитивна сторона осі у вказує на те, що стратегія відкритого коду є більш вигідною для автора, ніж стратегія пропрієтарного коду, тоді як негативна сторона осі у відображає, що стратегія пропрієтарного коду є більш вигідною.

Інші значення параметрів на рис.3.4: $s_{oo}^O = 10, s_{oc}^O = 8, , s_{cc}^O = 10, s_{co}^O = 1$ $s_{co}^P = s_{cc}^P = 0$ та $\beta_c = 5$.

Як зазначено вище, для розробника програмного забезпечення оптимально використовувати стратегію відкритого коду, коли витрати на розробку достатньо низькі, як показано на рис.3.4 в регіоні I. Цікавим є те, що автор обирає стратегію відкритого коду, коли зусилля контрибутора приносять йому більше користі, ніж самому контрибутору, тобто , $s_{\text{cc}}^{O} >$, s_{oc}^{O}. Фактично, у стані рівноваги контрибутор здійснює значні інвестиції в розвиток і стає лідером за якістю на ринку послуг, тоді як автор обслуговує нижній рівень ринку. Починаючи з низького β_{c}, рис.3.4 ілюструє, що зменшення ефективності контрибутора (тобто збільшення β_{c}) робить авторську стратегію кращою. Зокрема, зі збільшенням вартості розробки, як видно в регіоні II, в умовах політики відкритого коду ані автор не має стимулів інвестувати значні кошти на користь контрибутора, ані контрибутор не може отримати достатній дохід на ринку послуг. Натомість, контрибутор використовує свою економічну ефективність, щоб інвестувати в розробку до того моменту, коли його якісна пропозиція є досить привабливою порівняно з пропозицією автора, але недостатньо високою, щоб все ще існувала сильна цінова конкуренція, яка може зашкодити прибуткам. Тому стратегія відкритого коду є непривабливою для автора, і він обирає програмне забезпечення, яке залишається пропрієтарним.

Однак, коли ефективність контрибутора ще більше знижується, тобто вартість розробки β_{c} зростає, як це видно на рис.3.4 в регіоні III, стратегія відкритого коду може знову стати оптимальною. У цьому регіоні розробник може досягти достатньої диференціації як лідер якості, щоб зробити стратегію відкритого коду більш вигідною, ніж використання лише пропрієтарного коду, коли всі інвестиції спрямовуються на розробку пропрієтарного коду. Як наслідок, зниження ефективності контрибуторів робить політику відкритого коду оптимальною для розробника. Але, як обговорювалось вище, подальше збільшення β_{c} зменшує потенційне покращення якості від контрибутора в рамках стратегії відкритого коду і робить шлях пропрієтарного коду кращим для автора, як це також проілюстровано в регіоні IV на рис.3.4.

Все вище наведене дає уявлення про те, як економічна ефективність учасника впливає на стратегію розробника щодо вихідного коду.

4 ДОСЛІДЖЕННЯ МОДЕЛІ ОТРИМАННЯ ПРИБУТКУ ВІД РОЗРОБКИ ВІДКРИТИХ ПЛАТФОРМ

4.1 Опис моделі монетизації платформи з відкритим кодом

У цьому розділі описується модель для вивчення способів отримання прибутку фірми, що займається розробкою відкритого програмного забезпечення, яке являє собою платформу для інших розробників ПЗ. Тобто, використовуючи відкриту платформу сторонні розробники можуть створювати свої, навіть пропрієтарні, застосунки, однак мають дотримуватись вимог творців платформи, щодо розподілу прибутків від розповсюдження застосунків створених на давній платформі.

Способами поширення застосунків створених на платформі є продаж застосунку або безкоштовне поширення за умови інтеграції реклами в застосунок. Компанія розробник платформи контролює шляхи розповсюдження сторонніх застосунків на платформі і може диктувати умови для розповсюдження додатків.

Припускається, що ринок є монопольним, де немає конкуренції для фірми розробника платформи. У проєкті платформи з відкритим кодом фірма-розробник може спробувати мотивувати розробників додатків приєднатися до проєкту та розробляти більше додатків на платформу за допомогою грошових стимулів. Наприклад, щоб залучити більше розробників додатків, компанія може зменшити вимоги, щодо сплати відсоток від прибутку від розповсюдження створених додатків.

В модель закладається, що фірма розробник платформи намагається максимізувати свої прибутки контролюючи відсоток прибутку, який вона отримує від продажів розробників додатків, і впливаючи на відсоток платних (проти безкоштовних, що підтримуються рекламою) додатків. Витрати фірми з відкритим кодом на розробку платформи фірми з відкритим вихідним кодом на розробку платформи є фіксованими,, а змінні витрати відсутні. Модель

ігнорує фіксовані витрати, оскільки вони не відіграють жодної ролі у визначенні відсотка, що максимізує прибуток, та відсотка платних додатків.

4.1.1. Розміри мереж користувачів та компонентів

Як мережа компонентів (наприклад, кількість додатків доступних на платформі), так і мережа користувачів (наприклад, кількість користувачів платформою) відіграють важливу роль в успіху проєкту платформи з відкритим вихідним кодом. Тому, є доцільним проаналізувати як відсоток розподілу прибутку впливає на зростання кількості компонентів та мережі користувачів.

Нехай $0 \leq \theta \leq 1$ позначає відсоток розподілу прибутку фірми розробника відкритої платформи. Якщо фірма з відкритим кодом дозволяє розробникам залишати собі більшу частину прибутку, то більше розробників додатків будуть мотивовані приєднатися до платформи, що призводить до розширення мережі доповнень [23]. Припускається, що кількість розробників додатків зростає лінійно зі збільшенням відсотку від прибутку, який вони можуть отримувати. Якщо їх відсоток від прибутку дорівнює нулю, жоден розробник не бере участі у створенні застосунків на платформі. Крім того, припускається, що кількість додатків лінійно зростає зі збільшенням кількості розробників.

Позначимо Q_a як розмір мережі компонентів. Тоді маємо

$$Q_a = K(1 - \theta), \tag{4.1}$$

де $K > 0$ представляє максимальну кількість додатків, які розробники створили б, якби вони могли залишити собі весь прибуток, тобто $\theta = 0$.

Далі розглянемо вплив мережі доповнень на мережу користувачів. Нехай $0 \leq \xi \leq 1$ позначає відсоток платних додатків, а $1 - \xi$ - відсоток безкоштовних додатків, що підтримуються рекламою на платформі.

Ми припускаємо, що відсоток платних додатків надається фірмою з відкритим вихідним кодом розробникам додатків як частина стратегічного керівництва і не залежить від відсотка розподілу прибутку.

Як правило, платформа з більшою кількістю додатків може залучити більше користувачів [24]. Крім того, мережа користувачів зростає відносно повільно, коли всі додатки, доступні на платформі є платними і відповідно мережа користувачів зростає зі збільшенням кількості безкоштовних додатків [25]. Тут припускається, що розмір мережі користувачів зростає лінійно по відношенню до відсотка безкоштовних додатків, і що співвідношення між розміром мережі користувачів та розміром мережі доповнень задовольняє модель постійної еластичності [26].

Нехай N_o позначає розмір мережі користувачів, який визначається:

$$N_o = C(1 + \varepsilon - \xi)Q_a^x, \tag{4.2}$$

де C, ε та x - додатні константи. ε набагато менше 1, а C_ε відображає, якою мірою збільшення розміру мережі доповнення сприяє зростанню розміру мережі користувача при $\xi = 1$.

Константа x представляє еластичність розміру мережі користувачів до розміру мережі доповнення [27]. Вона може змінюватися в залежності від рівня насиченості ринку. Наприклад, коли кількість користувачів є відносно невелика і потенційний ринок покритий лише частково, очікується, що поява нових додатків може призвести до різкого збільшення кількості нових користувачів; таким чином, припускається, що $x \geq 1$. Коли потенційний ринок близький до насичення, очікується, що впровадження нових додатків призведе лише до помірного збільшення кількості нових користувачів; таким чином, припускається, що $0 \leq x < 1$. Нарешті, коли ринок насичений, очікується, що

поява нових додатків може не призвести до збільшення кількості нових користувачів; таким чином, x = 0.

Підставивши рівняння 4.1 в рівняння 4.2, маємо:

$$N_o = N_o(\theta, \varepsilon) = \alpha(1 + \varepsilon - \xi)(1 - \theta)^x, \quad (4.3)$$

де $\alpha = CK^x > 0$.

Рівняння 4.3 означає, що розмір користувацької мережі є максимальним, коли фірма з відкритим вихідним кодом не ділиться ніяким прибутку з розробниками додатків (тобто, θ = 0) і всі додатки на платформі стають безкоштовними
(тобто, $\xi = 0$).

Таким чином, щоб розширити мережу користувачів, фірма з розробник платформи повинна пообіцяти вищий відсоток від прибутку розробникам додатків та/або зменшити відсоток платних додатків на платформі.

4.1.2 Прибутки від продажу додатків

Дослідження прибутку фірми розробника платформи з відкритим вихідним кодом від платних додатків.

Нехай p_a позначає середню ціну платних додатків, а M_a^p позначає середню кількість платних додатків, які користувач завантажує. Тоді прибуток фірми з відкритим кодом фірми з відкритим кодом від додатків Π_1 можна обчислити за формулою:

$$\Pi_1 = \theta p_a M_a^p N_o, \quad (4.4)$$

Кількість додатків на одного користувача залежить як від середньої ціни продажу одного додатка [28] так і від загальної кількості платних додатків, доступних на ринку Android.

Наприклад, користувач телефону Android може встановити менше платних додатків, якщо середня ціна продажу одного додатка вища. І навпаки, якщо середня ціна продажу залишається незмінною, а платних додатків стає більше користувач може завантажити більше платних додатків, якщо середня ціна продажу залишається незмінною, оскільки для платформи доступно більше платних додатків.

Тут припускається, що зв'язок між середньою кількістю платних додатків на одного користувача та загальною кількістю платних додатків задовольняє функцію попиту з постійною еластичністю [26].

Тоді кількість платних додатків на одного користувача задається через:

$$M_a^p = f(p_a)(\xi Q_a)^y, \tag{4.5}$$

де $0 \leq y \leq 1$ - постійна еластичність, а ξQ_a представляє загальну кількість платних додатків на платформі.

Значення y може залежати від багатьох факторів, включаючи різноманітність доступних додатків та функціональності смартфона (наприклад, кількості додатків, які смартфон може вмістити для завантаження).

Функція $f(p_a)$ описує попит на платні додатки на одного користувача по відношенню до ціни p_a. Вона задовольняє умову $\frac{\partial f(p_a)}{\partial p_a} \leq 0$, тобто кількість платних додатків на одного користувача зменшиться, якщо середня ціна зросте. Якщо середня ціна p_a занадто висока, функція $f(p_a) = 0$, що означає, що користувачі не завантажують жодного платного додатку.

Підставивши рівняння 4.1, 4.2 та 4.5 у рівняння 4.4, отримаємо прибуток фірми з відкритим кодом від продажу платних додатків фірми з відкритим вихідним кодом від продажу платних додатків:

$$\Pi_1 = \Pi_1(p_a, \theta, \xi) = \mu p_a f(p_a)[\theta(1-\theta)^{x+y}\xi^y(1+\varepsilon-\xi)^x], \qquad (4.6)$$

де константа $\mu = CK^{x+y} > 0$.

Рівняння 4.6 показує, що фірма з відкритим кодом може отримувати прибуток від додатків тільки тоді, коли її частка прибутку $\theta \neq 0$ і $\theta \neq 1$, а частка платних додатків $\xi \neq 0$.

4.1.3. Прибуток від реклами в безкоштовних додатках

У цьому розділі аналізується прибуток фірми з відкритим вихідним кодом від реклами в додатках. У решті роботи, якщо не зазначено інше, використовується термін "реклама" для позначення реклами у додатку.

Припускається, що рекламодавці використовують мережу з відкритим вихідним кодом як рекламний носій для реклами в мобільних додатках [29], і що рекламодавці платять за кількість показів. Показ реклами визначається як завантаження веб-сторінки з рекламою (банером) на мобільному пристрої [30]. Вартість реклами, пов'язана з одним показом, зазвичай описується як CPM (вартість за показ) [30]. Нехай p_i позначає середню ціну за показ, N_a - загальну кількість рекламодавців на платформі, а M_i - середню кількість показів, яку рекламодавець хоче розмістити на платформі. Припускається, що є більше рекламних місць, ніж рекламодавці вимагають.

Тоді прибуток фірми з відкритим кодом від прибуток фірми з відкритим вихідним кодом від реклами Π_2 обчислюється за формулою:

$$\Pi_2 = \theta p_i M_i N_a, \tag{4.7}$$

При виборі рекламних носіїв рекламодавці, як правило, віддають перевагу мережі з більшою аудиторією; отже, чим більшою є мережа користувачів, тим привабливішою вона є для рекламодавців [31]. Крім того, рекламодавці менш схильні розміщувати рекламу в невеликій мережі з невеликою кількістю користувачів, і вони можуть почати розміщувати рекламу в мережі з відкритим вихідним кодом лише тоді, коли вона досягає певного розміру. Отже, кількість рекламодавців N_a визначається як [26]:

$$\mathrm{N}_a = \begin{cases} \gamma N_o^z, & \mathrm{N}_o \geq N_{o,min} \\ 0, & \mathrm{N}_o < N_{o,min} \end{cases}, \tag{4.8}$$

де $z > 0$ позначає константу еластичності, та константу $\gamma > 0$.

$N_{o,min}$ позначає мінімальний розмір мережі користувачів, який необхідний для того, щоб почати залучати рекламодавців. Припускається, що $N_{o,min}$ набагато менше ніж розмір мережі користувачів, який максимізує прибуток фірми з відкритим вихідним кодом від реклами.

Крім того, на середню кількість показів на одного рекламодавця M_i впливає середня ціна за показ p_i, середня кількість безкоштовних додатків на одного користувача M_a^p та розмір мережі користувачів N_o [31]. Вища ціна за показ призведе до вищої вартості реклами; таким чином, рекламодавці можуть не захотіти розміщувати рекламу в мережі з відкритим вихідним кодом, що призведе до меншої кількості показів. Крім того, зі збільшенням розміру мережі користувачів збільшується, мережа стає більш привабливою для рекламодавців як рекламний носій; таким чином, кількість показів на одного рекламодавця стає більшою (31). Тут припускається, що рекламодавці завжди хочуть підтримувати кількість показів на одного користувача. Тоді маємо

$$M_i = g(p_i)M_a^f N_o = g(p_i)(\,(1-\xi)Q_a)^y N_o, \tag{4.9}$$

де середня кількість безкоштовних додатків на одного користувача, M_a^f залежить лише від загальної кількості безкоштовних додатків, доступних на платформі.

Таким чином, слідуючи тим самим рядкам рівняння, маємо

$$M_a^f=[(1-\xi)Q_a\,]^y, \tag{4.10}$$

Функція $g(p_i)$ описує попит на покази для кожного рекламодавця по відношенню до ціни p_i. Вона задовольняє умову $\frac{\partial f(p_i)}{\partial p_i} \leq 0$, що означає, що кількість показів на одного рекламодавця зменшується, якщо середня ціна за показ зростає. Якщо середня ціна p_i занадто висока, функція $g(p_i) = 0$.

Підставивши рівняння 4.1, 4.2, 4.8 та 4.9 у рівняння 4.7, отримаємо прибуток фірми з відкритим вихідним кодом від реклами:

$$\Pi_2 = \Pi_2(\,p_i, \theta, \xi\,) = \begin{cases} \lambda p_i g(p_i)\theta(1-\theta)^{y+x(z+1)}(1-\xi)^y(1+\varepsilon-\xi)^{1+z}\ , & (\theta,\xi)\ \in S \\ 0, & (\theta,\xi)\ \notin S \end{cases}, \tag{4.11}$$

де константа $\lambda = \gamma C^{1+z}K^{x(1+z)+y} > 0$.

Тут S позначає можливу множину значень θ та ξ, таких, що результуючий розмір мережі користувачів не є менший за $N_{o,min}$, тобто

$$S = \{(\theta,\xi)\big|N_o(\theta,\xi) \geq N_{o,min}\}, \tag{4.12}$$

4.2 Практичне застосування моделі монетизації розробки платформ

4.2.1 Визначення ролі розміру мережі користувачів на прибуток від продажу

Збільшення розміру мережі користувачів не обов'язково збільшує прибуток фірми з відкритим вихідним кодом від додатків. Надмірне збільшення розміру мережі користувачів за рахунок зменшення відсотка розподілу прибутку та/або відсотка платних додатків призведе до зменшення прибутку від додатків. Доказ даного твердження наведений в додатку Г.

Розмір мережі користувачів загалом відіграє важливу роль у генеруванні прибутку від додатків. Щоб збільшити розмір мережі користувачів, фірма з відкритим вихідним кодом може зменшити свій відсоток від прибутку та/або зменшити відсоток платних додатків.

Перший спосіб збільшує розмір мережі користувачів, мотивуючи більше розробників додатків приєднатися до платформи, а отже, збільшує кількість додатків, доступних на платформі. Іншими словами, він розширює мережу користувачів за рахунок збільшення розміру мережі доповнень. На противагу цьому, остання збільшує розмір мережі користувачів за рахунок залучення більшої кількості користувачів, які приєднуються до мережі, оскільки стає доступною більша кількість безкоштовних додатків.

Хоча фірмі з відкритим вихідним кодом загалом вигідно збільшувати розмір мережі користувачів, зростання мережі користувачів понад певний розмір може зашкодити прибутку фірми з відкритим вихідним кодом від додатків. Доказ наведених вище тверджень в додатку Г надає оптимальний відсоток розподілу прибутку θ_1^* та оптимальний відсоток платних додатків ξ_1^*, що максимізують прибуток фірми з відкритим кодом від додатків, а також відповідний розмір мережі $N_{o,1}^*$. Зменшення частки прибутку фірми з відкритим кодом нижче θ_1^* все ще заохочує більше розробників додатків брати участь у проєкті з відкритим кодом, що призводть до збільшення розміру як

мережі доповнень, так і мережі користувачів. Однак, коли $\theta < \theta_1^*$, частка прибутку фірми з відкритим кодом є занадто низькою, щоб збільшити її прибутки. Насправді, негативний ефект від нижчого відсотка участі у прибутку переважає позитивний вплив більшої мережі користувачів на прибуток, що призводить до зниження прибутку. Аналогічно, зменшення відсотка платних додатків нижче ξ_1^* може привабити більше користувачів, оскільки з'являється більше безкоштовних додатків. Однак, без достатньої кількості платних додатків для продажу, ця стратегія шкодить прибуткам фірми з відкритим кодом від додатків.

Підсумовуючи, фірма розробник платформи з відкритим кодом може зазнати зниження прибутку від додатків через надмірне використання непримусової влади (тобто, надання занадто великої винагороди розробникам додатків) та/або через надмірне зниження відсотка платних додатків.

4.2.2 Максимізація прибутку за рахунок корегування мережі користувачів

Збільшення розміру мережі користувачів не обов'язково збільшує прибуток фірми з відкритим вихідним кодом від реклами. Надмірне збільшення розміру мережі користувачів за рахунок зменшення відсотку розподілу прибутку призведе до зменшення прибутку від реклами.

Розрахунки наведені в додатку Д передбачають, що максимізація розміру мережі користувачів не максимізує прибуток фірми з відкритим вихідним кодом від реклами. Доведення вищенаведеного твердження надає оптимальний відсоток розподілу прибутку θ_2^* та оптимальний відсоток платних додатків $\xi_2^* = 0$, які максимізують прибуток фірми з відкритим вихідним кодом від реклами, а також відповідний розмір мережі $N_{o,2}^*$. Тут однією з умов максимізації прибутку фірми з відкритим вихідним кодом від

реклами є те, що всі додатки на платформі повинні бути безкоштовними (тобто, $\xi_2^* = 0$). Як тільки всі додатки стануть безкоштовними, для подальшого зростання мережі користувачів фірма з відкритим вихідним кодом повинна зменшити власний відсоток від прибутку нижче θ_2^*. Зменшення частки прибутку фірми з відкритим вихідним кодом заохочує більше розробників додатків брати участь у проєкті з відкритим вихідним кодом, збільшує кількість доступних додатків і, як наслідок, збільшує розмір мережі користувачів. Однак, негативний ефект від нижчого відсотку участі у прибутку переважає позитивний ефект від більшої мережі користувачів, зменшуючи прибуток від реклами.

4.2.3 Визначення оптимального розподілу прибутку від продажів додатків та реклами

Розподіл прибутку може бути заданий наступними твердженнями.

По-перше, Оптимальний відсоток розподілу прибутку, що максимізує прибуток фірми з відкритим вихідним кодом від реклами є нижчим, ніж той, що максимізує прибуток від додатків.

По-друге, Для максимізації прибутку від реклами фірмі з відкритим кодом потрібна більша мережа користувачів фірмі з відкритим кодом для максимізації прибутку від реклами потрібна більша мережа користувачів, ніж для щоб максимізувати прибуток від реклами, ніж для максимізації прибутку від додатків.

Доказ наведених тверджень знаходиться в додатку Е. Твердження вказує на те, що оптимальний відсоток розподілу прибутку θ_1^* який максимізує прибуток фірми з відкритим кодом від додатків не максимізує її прибуток від реклами. Для того, щоб максимізувати прибуток від реклами, фірма з відкритим кодом повинна знизити свій відсоток розподілу прибутку нижче θ_1

до θ_2^* і ще більше збільшити розмір мережі користувачів. Іншими словами, рекламний бізнес потребує більшої мережі для максимізації прибутку, ніж бізнес додатків.

Крім того, твердження у другому абзаці припускає, що збільшення розміру мережі має бути більш важливим стратегічним фактором для успіху рекламного бізнесу, ніж для успіху бізнесу додатків. У підсумку, це означає, що використання непримусової влади (наприклад, винагороди для розробників додатків) є більш важливіше для рекламного бізнесу, ніж для бізнесу додатків. Для того, щоб максимізувати свій прибуток від реклами, фірма з відкритим вихідним кодом повинна використовувати непримусову владу, щоб мотивувати більшу кількість розробників додатків приєднатися до платформи приєднатися до платформи та ще більше збільшити розмір мережі. Крім того, вибір відсотка розподілу прибутку може також залежати від стратегічного пріоритету фірми з відкритим вихідним кодом щодо додатків чи реклами бізнесу.

Наприклад, якщо пріоритетним є рекламний (прикладний) бізнес над прикладним (рекламним) бізнесом, фірма з відкритим кодом може вибрати загальний відсоток розподілу прибутку $\theta_2^* < \theta < \theta_1^*$ але близький до θ_2^* (θ_1^*), навіть жертвуючи прибутком від його застосування (реклами) бізнесу.

4.2.4 Візуалізація розподілу прибутку та розміру мережі користувачів

Для доповнення, наводяться числові приклади з $x = y = z = 1$, де ілюструється вплив відсотка розподілу прибутку θ та відсотка платних додатків ξ на прибуток фірми з відкритим вихідним кодом в залежності від розміру мережі користувачів. Параметри x, y та z можуть впливати на величину прибутку, але вони не впливають на висновки щодо максимізації прибутку. На рис.4.1; рис.4.2 та рис.4.3 показано вплив відсотка розподілу

прибутку θ на прибуток $П$ для різних відсотків платних додатків ξ. рис.4.1 зображає залежність на прибутки від додатків, в той час коли рис.4.2 відображає залежність на прибутки від реклами. Символ "■" позначає точку $(\theta^*, \xi, П)$. Вісь Y відкладає відносний прибуток до максимального прибутку від додатків або реклами (наприклад, $П_1 = 0.5$ - половина максимального прибутку).

По-перше, спостерігається вплив відсотка розподілу прибутку на прибуток від додатків. Як показано на рис.4.1, для фіксованого відсотка платних заявок ξ прибуток $П_1$ зростає зі зменшенням відсотка участі в прибутку θ, і досягає максимального значення при $\theta = \theta_2^*$ ($\theta = \theta_1^*$).

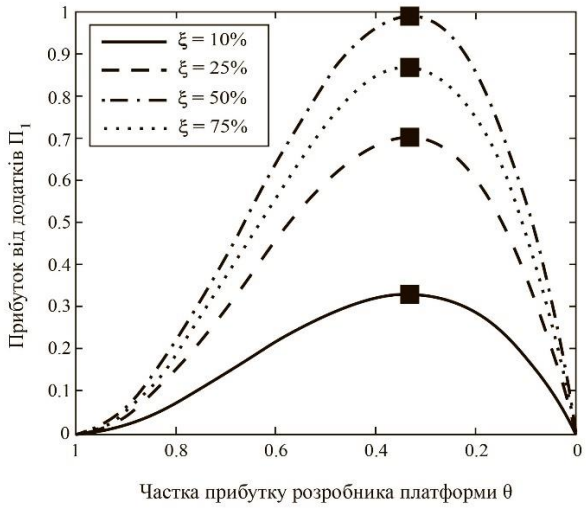

Рисунок 4.1 – Залежність розподілу прибутку θ на прибуток від додатків $П_1$

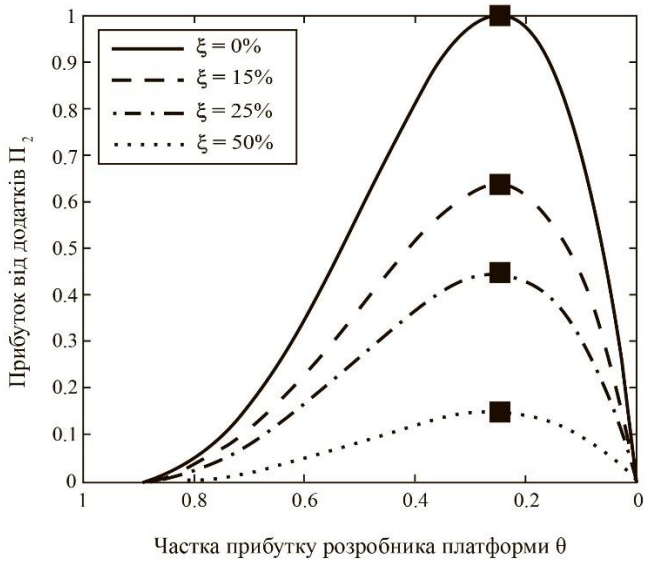

Рисунок 4.2 – Залежність розподілу прибутку θ на прибуток від додатків $П_2$

Подальше зменшення частки прибутку θ нижче θ_2^* (θ_1^*) призводить до зменшення прибутку $П_1$ ($П_2$). Порівняння рис.4.1 і рис.4.2 показує, що в той час як прибуток $П_1$ максимізується при $(\theta_1^*, \xi_1^*) = (\frac{1}{3}, \frac{1+\varepsilon}{2})$, прибуток $П_2$ максимізується при $(\theta_2^*, \xi_2^*) = (\frac{1}{4}, 0)$, таким чином, $\theta_1^* \neq \theta_2^*$і $\xi_1^* \neq \xi_2^*$.

Далі розглядається залежність між розміром мережі користувачів та прибутком від додатків та реклами на рис.4.3 та рис.4.4. Для заданого відсотка платних додатків ξ прибуток $П_1$ ($П_2$). продовжує зростати зі збільшенням мережі до $N_o(\theta_1^*, \xi)$ для прибутку від додатків і $N_o(\theta_2^*, \xi)$ для прибутку від реклами. Символ "●" відповідає $(N_o(\theta^*, \xi), П)$. Для заданого відсотка платних додатків ξ, збільшення розміру мережі користувачів N_o відповідає зменшенню відсотка розподілу прибутку θ. Розмір мережі користувачів являє собою відношення розміру мережі до максимального розміру мережі (наприклад, $N_o = 0.5$ означає половину максимального розміру мережі).

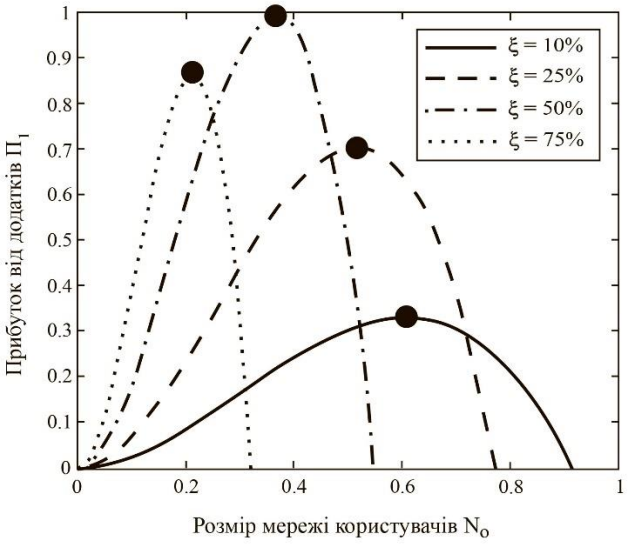

Рисунок 4.3 – Вплив мережі користувачів N_o на прибуток від додатків $П_1$

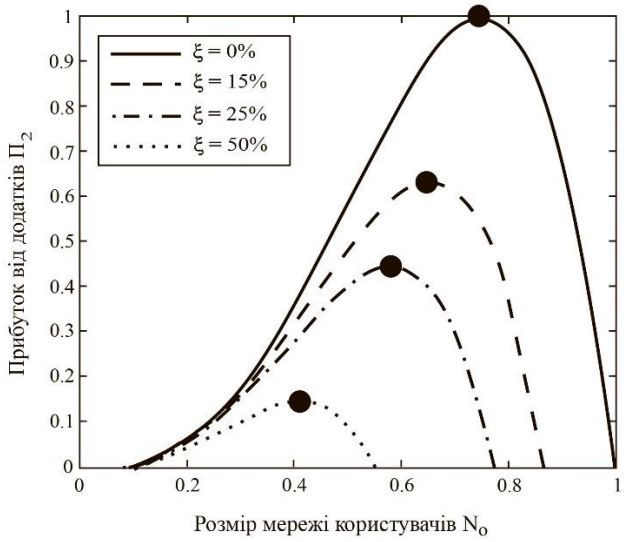

Рисунок 4.4 – Вплив мережі користувачів N_o на прибуток від реклами $П_2$

Однак подальше зростання мережі зменшує прибуток $П_1$ ($П_2$). Як показано на рис.4.3, підтримуючи відносно високий відсоток платних додатків (наприклад, $\xi = 75\%$), фірма з відкритим кодом може стикнутися з високою чутливістю прибутку $П_1$ до розміру мережі. Таким чином, фірма з відкритим кодом може відносно швидко отримати прибуток від додатків, навіть якщо мережа користувачів є відносно невеликою.

Однак, відносно високий відсоток платних додатків може перешкоджати зростанню мережі та прибутку від додатків. Таким чином, відсоток платних додатків повинен бути знижений ($\xi = 50\%$) для подальшого зростання розміру мережі і збільшення прибутку. Однак, надмірне зменшення відсотка платних додатків зменшує чутливість прибутку Π_1 до розміру мережі. Наприклад, при $\xi = 25\%$ прибуток Π_1 зростає набагато повільніше по відношенню до розміру мережі користувачів, ніж при $\xi = 50\%$ або $\xi = 75\%$. Крім того, як показано на рис. 2b, коли відсоток платних додатків є відносно високим (наприклад, $\xi = 50\%$), прибуток від реклами Π_2 зростає відносно повільно по мірі розширення мережі користувачів. Коли фірма з відкритим вихідним кодом зменшує відсоток платних додатків, прибуток від реклами стає більш чутливим до розміру мережі. Коли всі додатки на платформі безкоштовні (тобто $\xi = 0$), прибуток від реклами максимізується при розмірі мережі $N_o(\theta_2^*, 0)$.

Для подальшого розширення мережі користувачів фірма з відкритим вихідним кодом повинна знизити свою частку прибутку нижче θ_2^*. Однак така стратегія призводить до зменшення прибутку від реклами Π_2. Порівнюючи рис.4.3; рис.4.4, розмір мережі, яка максимізує прибуток від реклами, набагато більший, ніж та, яка максимізує прибуток від додатків. Це означає, що бізнес додатків дозволяє фірмі максимізувати прибуток навіть тоді, коли мережа ще відносно невелика. Коли мережа зростає, фірма з відкритим вихідним кодом може скористатися перевагами своєї більшої мережі, щоб максимізувати прибуток від реклами, а не прибуток від додатків.

ВИСНОВКИ

Під час виконання магістерської роботи було проаналізована предметну галузь розробки ІТ-проєктів з відкритим кодом та існуючі бізнес-моделі монетизації ІТ-проєктів, було детально досліджено моделі отримання прибутку від консультаційних послуг та моделі отримання прибутку від розробки відкритих платформ, описано практичне застосування досліджуваних моделі монетизацій.

Під час аналізу існуючих методів вирішення поставленої задачі було наведено загальний опис існуючих бізнес-моделей монетизації ІТ-проєктів з відкритим кодом.

Аналіз отримання прибутку від консультаційних послуга щодо ІТ-проєктів з відкритим кодом був математично змодельований та застосований до вирішення практичних задач. Наприклад, задача визначення доцільності застосування стратегії відкритого коду в порівнянні з пропрієтарною стратегію, виходячи з витрат та отриманого прибутку.

Наступною проаналізованою моделлю монетизації є отримання прибутку від розробки відкритих платформ. Модель була описана та застосована для розрахунку прибутковості від різних способів розповсюдження компонентів платформи, а також вплив мережі користувачів на прибутковість різних типів розповсюдження компонентів. Практичні задачі вирішувались на прикладі платформи розповсюдження додатків, що могли бути платними(за типом розповсюдження), або безкоштовними, але містили рекламу, що генерувала прибуток, як для розробника додатку, так і розробника відкритої платформи.

Проаналізовані моделі цінні як у плануванні, так і підтримці ІТ-проєктів з відкритим кодом, оскільки описують різні типи проєктів на різних стадіях розробки з огляду на їх прибутковість, а також спираючись на особливості розробки відкритих ІТ-проєктів як таких.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проєктами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 28 с.

2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

3. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.

4. Linux Foundation, Enterprise End User Report Linux Adoption: Third Annual Survey of World's Largest Enterprise Linux Users. The Linux Foundation; 2013.

5. Noyes K., Security and quality top companies' reasons for using open source. PCWorld; 2014.

6. Herz J., M. Lucas, J. Scott, Open technology development: Roadmap plan. U.S. Department of Defense; 2006.

7. Glick B., Government mandates 'preference' for open source. ComputerWeekly; 2013.

8. Lakshminarayanan R., Open Source provides compelling benefits to business. ZDNet; 2014.

9. McMillan R., Red Hat becomes open source's first $1 billion baby. Wired; 2012.

10. Holmstrom B., Milgrom P., Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design. Journal of Law Economics, & Organization 7; 1991. – С. 24–52.

11. Fishman S., Open-source licenses are not all the same. O'Reilly; 2004.

12. Hillesley R., Red Hat at the crossroads. ITPRO; 2008.

13. Rusin Z., Open Source Software license information; 2008. URL: https://www.vacuubrand.com/context/other_documents/Open_Source_Software_license_information.pdf (дата звернення: 25.03.2023).

14. Montague B., Why you should use a BSD style license for your open-source project. FreeBSD.org; 2008.

15. Stewart K. J., A. P. Ammeter, L. M. Maruping. Impacts of license choice and organizational sponsorship on user interest and development activity in open-source software projects. InformationSystems Research 17 (2); 2006. – С. 126–144.

16. Lerner, J. and J. Tirole, The scope of open-source licensing. Journal of Law, Economics, & Organization 21 (1); 2005. – С. 20–56.

17. Osterwalder A, The business model ontology - a proposition in a design science approach, University of Lausanne, Switzerland.; 2004. – С. 22-26.

18. Hitt, L. M., D. J. Wu, X. Zhou, Investment in enterprise resource planning: Business impact and productivity measures. Journal of Management Information Systems 19 (1).; 2002. – С. 71–98.

19. Spiller, D. and T. Wichmann, FLOSS 3: Basics of Open-Source Software Markets and Business Models, in FLOSS - Free/Libre Open-Source Software: Survey and Study. Berlecon Research: Berlin; 2002. – С.12-20.

20. Osterwalder A, Pigneur Y., Tucci C. Clarifying Business Models: Origins, present and Future of the Concept, CAIS; 2005. – 16 с.

21. Raymond, E. S., The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary (1 ed.). Sebastopol, CA: O'Reilly & Associates, Inc; 1999.

22. Oi, W. Y., Heterogeneous firms and the organization of production. Economic Inquiry 21 (2); 1983. – С.147–171.

23. Coar, K., The open-source definition (annotated); 2006. URL: http://www.opensource.org/docs/definition.php (дата звернення: 01.04.2023).

24. Frels, J. K., Shervani, T., Srivastava, R. K., The integrated networks model: Explaining resource allocations in network markets. Journal of Marketing, 67(1); 2003.– C.29–45.

25. Tibken, S., Google ties Apple with 700,000 Android apps. Cnet; 2012. URL: http://news.cnet.com/8301-1035_3-57542502-94/google-ties-apple-with-700000-android-apps/(дата звернення: 06.04.2023).

26. Manoogian, J., How free apps can make more money than paid apps. TechCrunch; 2012. URL: http://techcrunch.com/2012/08/26/how-free-apps-can-make-more-money-than-paid-apps/(дата звернення: 11.04.2023).

27. Coffey, W., Allometric growth in urban and regional social-economic systems. Canadian Journal of Regional Science, 79(1); 1979. – C. 49–65.

28. Lambin, J., Is gasoline advertising justified? Journal of Business, 45; 1972. – C. 585–619.

29. Austin, S., The surprising numbers behind apps. Wall Street Journal Digits; 2013. URL: https://www.wsj.com/articles/BL-DGB-26360 (дата звернення: 14.04.2023).

30. Infiniti Research Limited. Global mobile ad spending market 2012–2016. Research and markets; 2013. URL: https://www.prnewswire.com/news-releases/global-mobile-application-market-2012-2016-228930891.html (дата звернення: 14.04.2023).

31. Ahmed, M. T., Kwon, C., Pricing game of online display advertisement publishers. European Journal of Operational Research, 219(2); 2012. – C. 477–487.

32. Gabszewicz, J. J., Laussel, D., Sonnac, N., Programming and advertising competition in the broadcasting industry. Journal of Economics & Management Strategy, 13(4); 2004. – C. 657–669.