

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 моделей і методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків»
(тема)

Виконав:
студент 2 курсу, групи УПГІТМ-20-1
Ярмак В.В
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва освітньої програми)

Керівник проф.каф.ІУС Євланов М.В
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ проф. Петров К.Е
(підпис) (прізвище, ініціали)

2021 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-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«__» _____ 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові _____ Ярмаку Валерію Вячеславовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження моделей і методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків».

Затверджена наказом університету від 05.11.2021р. № 1646 Ст

2. Термін подання студентом роботи до екзаменаційної комісії: 13.12.2021р.

3. Вихідні дані до роботи: науково-технічні публікації та інтернет-джерела з тематики кваліфікаційної роботи

4. Перелік питань, що потрібно опрацювати в роботі: формування проблеми щодо вдосконалення; аналіз існуючих методів вирішення проблеми оцінювання трудовитрат ІТ-проектів з розробки мобільних додатків; вибір цілі та напрями удосконалення; постановка завдань до вирішення даної проблеми; розробка модифікованого методу оцінки трудовитрат ІТ-проекту з розробки мобільного додатку; розробка етапів оцінювання вартості ІТ-проекту з використанням модифікованого методу; розробка алгоритму реалізації модифікованого методу; розробка інструкції членам команди з виконанням розробленого методу; апробація методу під час оцінювання трудовитрат ІТ-проекту з розробки мобільного додатку.

КАЛЕНДАРНИЙ ПЛАН

№ №	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літератури та інтернет джерел	01.11.2021– 05.11.2021	Виконано
2	Опис постановки задачі дослідження	06.11.2021 – 07.11.2021	Виконано
3	Обробка матеріалів передатестаційної практики	08.11.2021 – 10.11.2021	Виконано
4	Аналіз існуючих методів оцінки трудовитрат з розробки мобільних додатків	11.11.2021– 13.11.2021	Виконано
5	Розробка модифікованого методу «секторів»	14.11.2021 – 19.11.2021	Виконано
6	Опис практичного вирішення задачі оцінювання трудовитрат розробки ІТ-проекту з розробки мобільного додатку	14.11.2021 – 19.11.2021	Виконано
7	Апробація методу секторів під час оцінювання трудовитрат ІТ-проекту з розробки мобільного додатку	20.11.2021– 25.11.2021	Виконано
8	Підготовка та оформлення пояснювальної записки та графічного матеріалу до кваліфікаційної роботи	20.11.2021– 25.11.2021	Виконано
9	Підготовка презентаційних матеріалів	26.11.2021– 27.11.2021	Виконано
10	Перевірка кваліфікаційної роботи на норм-контроль	06.12.2021	Виконано
10	Перевірка кваліфікаційної роботи на плагіат	08.12.2021	Виконано
11	Попередній захист кваліфікаційної роботи	10.12.2021	Виконано
12	Захист кваліфікаційної роботи	13.12.2021	Виконано

Дата видачі завдання 08.11.2021р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Євланов М.В
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 104 стор., 63 рисунків., 8 табл., 1 додаток, 17 джерел.

ДОДАТОК, КАЛЕНДАРНИЙ ПЛАН, МОБІЛЬНИЙ, МОДЕЛЬ, МОДИФІКАЦІЯ МЕТОДУ, ПРОЕКТ, ПРОФОРІЄНТАЦІЯ, ОЦІНКА ТРУДОВИТРАТ, СПИСОК РОБІТ.

Актуальність цієї роботи проявляється в необхідності забезпечення таким методом або моделлю, який надав(-ла) би можливість більш ефективно та легко оцінювати трудовитрати ІТ-проекту з розробки мобільних додатків ніж існуючі моделі або методи.

Об'єктом дослідження кваліфікаційної роботи є процес оцінювання трудовитрат ІТ-проекту з розробки мобільних додатків.

Метою досліджень кваліфікаційної роботи є дослідження моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків, та розробка або модифікація моделі або методу оцінки даної задачі, який буде більш ефективний, ніж існуючі моделі або методи та легкий у використанні.

Методи дослідження: аналіз та синтез схожих інтуїтивних методів оцінки трудовитрат, використання деяких їх принципів за аналогією.

Отримані наукові результати: розроблено модифікований метод оцінки трудовитрат ІТ-проекту з розробки мобільного додатку; запропоновано алгоритм реалізації модифікованого методу; запропоновано опис використання модифікованого методу командою, запропонована апробація модифікованого методу.

Результати кваліфікаційної роботи можуть бути впроваджені в діяльність ІТ-компаній, які займаються розробкою мобільних додатків й не тільки.

Проведена апробація модифікованого методу «Секторів» на мобільному додатку з профорієнтації показала, що даний метод дійсно дозволяє більш ефективно та легко визначити оцінку трудовитрат ІТ-проекту з розробки мобільних додатків ніж існуючі моделі або методи.

ABSTRACT

The explanatory note to the attestation work contains: 104 pages, 63 figures., 1 attachment, 8 tables., 17 sources.

APPENDIX, CALENDAR PLAN, MOBILE, MODEL, MODIFICATION OF THE METHOD, PROJECT, PROFESSIONAL TRAINING, ESTIMATION OF LABOR COSTS, LIST OF WORKS.

The relevance of this work is manifested in the need to provide a method or model that would provide an opportunity to more effectively and easily assess the labor costs of an IT project to develop mobile applications than existing models or methods.

The object of research of the master's qualification work is the process of estimating the labor costs of an IT project for the development of mobile applications.

The purpose of the master's thesis is to study models and methods of estimating labor costs of an IT project for mobile application development, and to develop or modify a model or method of evaluation of this task that will be more effective than existing models or methods and easy to use.

Research methods: analysis and synthesis of similar intuitive methods of labor cost estimation, use of some of their principles by analogy.

Scientific results obtained: a modified method for estimating the labor costs of an IT project for the development of a mobile application was developed; the algorithm of realization of the modified method is offered; the description of use of the modified method by the command is offered, the approbation of the modified method is offered.

The results of the master's qualification work can be implemented in the activities of IT companies engaged in the development of mobile applications and more.

The approbation of the modified Sectors method on the mobile career guidance application showed that this method really allows more efficient and easy determination of the cost estimate of an IT project for mobile application development than existing models or methods.

ЗМІСТ

Скорочення та умовні позначки	10
Вступ.....	11
1 Аналіз існуючих моделей та методів оцінки трудовитрат з розробки мобільних додатків	12
1.2 Дослідження типів мобільних додатків.....	15
1.2.1 Нативні додатки.....	15
1.2.2 Мобільні веб-додатки	16
1.2.3 Гібридні додатки	16
1.3 Аналіз основних моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків	17
1.3.1 Аналіз основних інтуїтивних моделей та методів оцінки трудовитрат ІТ проекту з розробки мобільних додатків	17
1.3.2 Аналіз основних формальних моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків.....	24
1.4 Аналіз існуючих інформаційних систем моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків.....	26
1.4.1 Аналіз існуючих інформаційних систем формальних моделей та методів оцінки трудовитрат ІТ проекту з розробки мобільних додатків.....	26
1.4.2 Аналіз існуючих інформаційних систем інтуїтивних моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків.....	29
1.5 Формування мети та постановка задачі кваліфікаційної роботи.....	32
2 Розробка модифікованого методу «секторів»	33
2.1 Синтез модифікованого методу «Секторів» за допомогою інтеграцією методів «Big/small/Uncertain» , «Maximum Size or Less» та «Ordering Rule».....	33

2.2	Етапи оцінювання вартості ІТ-проекту з використанням модифікованого методу	34
2.3	Модифікація методів «Big/small/Uncertain», «Max Size or Less»	39
2.3.1	Використання методу «Big/small/Uncertain» для модифікації	39
2.3.2	Використання методу «Maximum Size or Less» для модифікації.....	40
2.3.3	Використання методу «Ordering Rule» для модифікації.....	41
3	Опис практичного вирішення задачі оцінювання трудовитрат розробки ІТ-проекту з розробки мобільного додатку.....	43
3.1	Розробка алгоритму реалізації модифікованого методу оцінювання трудовитрат ІТ-проекту з розробки програмного додатку.....	43
3.2	Розробка інструкції команді з виконанням розробленого методу....	46
4	Апробація методу секторів під час оцінювання трудовитрат ІТ-проекту з розробки мобільного додатку.....	49
4.1	Статут проекту	54
4.2	Структуризація проекту	57
4.3	Керування часом, вартістю та ресурсами проекту	62
4.3.1	Розробка моделі проекту за допомогою інформаційної технології MS Project	62
4.3.2	Оптимізація проекту за показниками час-вартість.....	65
4.3.3	Планування системи управління проектом.....	67
4.4	Приклад використання модифікованого методу оцінки трудовитрат ІТ-проекту з розробки мобільного додатку.....	72
	Висновки	82
	Перелік джерел посилання.....	84
	Додаток А. Графічний матеріал	86

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – портативний комп'ютер;

Agile – гнучка методологія розробки Agile;

Android – мобільна операційна система Android;

COCOMO – COnstructive COst MOdel;

Flutter – мова програмування Flutter;

GitHub – сервіс для хостинга ІТ проектів;

Google App – сервіс для скачування додатків на Android;

Google Store – сервіс для скачування додатків на IOS;

HTML – HyperText Markup Language;

IOS – мобільна операційна система IOS;

JIRA – система відстеження помилок;

Kanban - гнучкий підхід Kanban;

Kotlin – мова програмування Kotlin;

OBS – Organisation breakdown structure;

QA – Quality Assurance;

Scrum – гнучкий підхід Scrum;

UX – user experience;

WorkSection – система управління проектами.

ВСТУП

У нашому столітті інформаційних технологій, створюється величезна кількість ІТ-проектів: розробка веб-сайтів, ігор, системних утиліт, мобільних додатків і т.д. Для створення кожного з цих проектів необхідно придумати його ідею і концепцію, створити команду фахівців, написати технічні вимоги, налагодити бізнес-процеси всередині команди, розробити продукт, протестувати, впровадити його і т.д. Під час процесу розробки проекту можуть виникнути безліч підводних каменів і змін: замовник вирішив додати новий функціонал, розробка окремого модуля виявилася набагато складніше, ніж оцінювалася спочатку, і безліч інших ризиків.

Як видно, процес розробки ІТ-проекту є комплексним і досить складним. Тому, дуже важливо якомога точніше оцінити трудовитрати ІТ-проекту з розробки, щоб у компанії вистачило ресурсів для успішного його завершення. Необ'єктивна оцінка трудовитрат ІТ-проекту може привести до марної трати грошових ресурсів, а також до зайвих трудовитрат і виснаження команди проекту, що в свою чергу може збільшити ризик на банкрутство компанії і її подальше закриття.

У зв'язку з відсутністю ефективних методів чи моделей оцінювання трудовитрат ІТ-проекту з розробки мобільних додатків виникло завдання в усуненні цієї проблеми.

Саме тому, головною метою даної роботи є створення або модифікація ефективної(-го) моделі або методу оцінки трудовитрат ІТ-проекту з розробки мобільного додатка, який буде більш ефективний ніж існуючі моделі або методи, доволі швидкий, та легкий у використанні.

Робота виконується відповідно до вимог методичних вказівок [1].

Для оформлення даного звіту використовувався державний стандарт України [2].

1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ТА МЕТОДІВ ОЦІНКИ ТРУДОВИТРАТ З РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ

1.1 Аналіз відмінностей мобільної розробки проектів від десктопної розробки

Мобільну розробку програми можна винести в окремий підклас, оскільки даний вид розробки має свої відмінні риси. Розглянемо дані особливості з точки зору управління проектами.

Організаційна структура команди (OBS). Зараз кількість мов програмування для мобільного розробки бурхливо зростає. Зокрема, недавно вийшла кросплатформена мова програмування Flutter. Тому, проектному менеджеру необхідно добре подумати перед тим, як формувати команду, необхідно визначитися з вибором членів своєї команди: запросити нативних розробників – Kotlin, Android та IOS, або кросплатформених – Flutter. Більш того, проектний менеджер повинен враховувати, що найчастіше в малих і середніх компаніях в відділі мобільного розробки немає окремої людини, яка відповідає за бізнес-аналіз. Дану зону відповідальності ділить між собою UX дизайнер і безпосередньо лідер команди. Також, проектному менеджеру для розробки мобільних додатків доцільніше найняти саме manual QA, ніж Automation QA, так як немає необхідності в написанні програмних тестів, оскільки річний підхід швидше і простіше.

Список задач. Проектний менеджер повинен враховувати, що список завдань при розробці мобільного додатку може відрізнитися від списку завдань десктопного додатка (сайту). Наприклад, якщо взяти список завдань manual QA, то він має свою специфіку тестування для мобільних додатків, що в свою чергу відображається на списку завдань проекту (WBS). Наприклад, тестувальникам необхідно тестувати програми на великій кількості пристроїв з різною діагоналлю екрану і різним форм-фактором (планшети, смартфони і т.д).

Апаратні обмеження. Проектному менеджеру необхідно враховувати апаратні обмеження мобільного пристрою. Наприклад, одним з важливих параметрів є енергоспоживання мобільного додатка та пристрою в цілому.

Продуктивність. Незважаючи на те, що мобільні пристрої стають з кожним роком все потужніші, комп'ютери все одно випереджають за цим критерієм. Тому, важливу роль при розробці мобільного додатку грає оптимізація існуючої апаратної частини. Велика частина обчислювальної логіки, полягає в взаємодії з Інтернетом і з хмарними сервісами.

Формування розкладу. Тривалість життєвого циклу мобільного додатка зазвичай значно менше, ніж розробка програми для ПК. Даний момент проектний менеджер повинен враховувати при формуванні календаря і розмові з замовником.

Також, зазвичай версії мобільного застосування випускаються частіше в реліз, ніж десктопні. Це необхідно для того, щоб спробувати новий функціонал і подивитися на відгуки клієнтів. Це називається А / В тестуванням. Причина в тому, що отримати відгук користувачів на мобільних додатках можна швидше, ніж на десктопних, оскільки мобільний телефон завжди під рукою, а доступ до скачування з Google App або App Store простий і швидкий. Лідер команди повинен мати на увазі, що так як життєвий цикл розробки мобільного застосування коротше, ніж у десктопного, то кількість демо-днів і отримання відгуків відбувається частіше, оскільки швидше розробляється необхідний функціонал, ніж у десктопного.

Формування технічної документації. Досить часто в ІТ-компаніях проектним менеджером розписується саме список User Stories, UI / UX дизайнером малюється макети екранів додатка, а повноцінне технічне завдання не формується, оскільки в цьому немає необхідності, так як повноцінне технічне завдання вимагає великої кількості витраченого часу і фінансів.

Використання спеціалізованого ПЗ. В ІТ компаніях використовують тільки ті інструменти, які добре підходять для розробки конкретного виду програми. Наприклад, багато компаній використовують таку систему управління проектом, як JIRA. Але, використання даної програми є не зовсім доцільним для розробки мобільних додатків, оскільки саме цей процес не вимагає такої великої кількості налаштувань кастомізації системи управління проектом. Тому, часто використовується замість цього сервісу WorkSection .

Аналіз можливих ризиків. Проектний менеджер повинен враховувати, що оновлення апаратного забезпечення на мобільних пристроях відбувається більш динамічно, відповідно, виходить більша кількість пристроїв за один проміжок часу. Таким чином, ймовірність ризиків можливих помилок на мобільних пристроях вище.

Розподіл завдань між виконавцями. Для розробки мобільного застосування часто використовуються гнучкі методології Agile, зокрема, фреймворки Scrum і Kanban. Тому команди зазвичай крос-функціональні. Таким чином, в разі необхідності, іноді проектний менеджер може призначати деякі види завдань на виконання тим. членам команди, у яких інша спеціалізація

Графічний дизайн. При розробці мобільного додатку необхідно використовувати такий підхід, який засновано на принципах human-centered design тобто орієнтований на проблеми користувача. Необхідно робити акцент на інтерфейс [4].

Технічне завдання. На етапі проектування технічного завдання для мобільного додатка воно не записується текстом, а зображується у вигляді графічного матеріалу, який сприймати і реалізовувати набагато легше [4].

Апаратні можливості. При розробці мобільного додатку проектному менеджеру необхідно враховувати і той факт, що є можливість використовувати додатковий функціонал, на відміну від комп'ютера, а саме - GPS-датчик, акселерометр, компас, список контактів і все інше [4].

Вартість проекту. Зазвичай, вартість розробки мобільного застосування перевищує вартість розробки сайту, при цьому, найчастіше, сайти окупаються швидше [4].

Таким чином, ми бачимо, що з точки зору проектного менеджера, між розробкою десктопного додатка і мобільного є досить велика різниця. Таким чином, процес розробки мобільних додатків ми можемо виділити в окремий підклас.

1.2 Дослідження типів мобільних додатків

1.2.1 Нативні додатки

Даний вид додатків називається нативним, оскільки вони розробляються на рідній мові певної платформи. Наприклад, для платформи Android додаток – на мові Java. Для платформи IOS – на мові objective-C або Swift. Нативні додатки знаходяться на самому пристрої. Завантажити їх можна через відповідні магазини додатків, для Android – Play Market, для IOS – App Store. Так як вони спеціально розроблені й оптимізовані під певну платформу, то цей вид додатку буде мати високу продуктивність, а також, частково або повністю, даний вид додатків може обходитися без підключення до мережі Інтернет [5].

До плюсів нативного додатку ще можна віднести високу ступінь безпеки, розширений інтерфейс і зручність для кінцевого користувача. До мінусів нативних додатків можна віднести тривалі терміни розробки і необхідність в обхваті платформ [5].

1.2.2 Мобільні веб-додатки

Мобільні веб-додатки – це додатки, яке є насправді веб-сайтами під виглядом додатка, але при використанні можуть виглядати, як нативний додаток. Щоб скористатися мобільним веб-додатком необхідно мати доступ до Інтернету. Завдяки Інтернету відбувається оновлення даних в додатку. Запустивши веб-додаток користувач виконує ті ж дії, що і при в вході на будь-який веб-сайт. Тільки при цьому, він має можливість поставити іконку веб-додатка на свій робочий стіл мобільного пристрою [5].

Веб-додаток є кросплатформним і він не прив'язаний до конкретної платформи. А через те, що веб-додаток є мобільною версією сайту, то воно не займає місце в сховищі мобільного пристрою. Таким чином, з плюсів веб-додатку можна виділити повне охоплення платформ, простий і швидкий процес розробки (оскільки існує величезна кількість фреймворків), відсутність необхідності установки з магазину веб-додатків. До недоліків можна віднести обов'язкове підключення до Інтернету, спрощений інтерфейс програми, більш низька продуктивність і рівень безпеки програми в порівнянні з нативним, неможливість відправити push-повідомлення [5].

1.2.3 Гібридні додатки

Гібридні додатки – симбіоз нативного і веб-додатку. Аналогічно, як і нативний додаток, його можна завантажити в магазин додатків, плюс до цього, у цьому виду додатків можна використовувати функції пристрою, на яке воно встановлено. Однак, насправді, основою для гібридного додатку, як і для веб-додатку, є HTML5. Воно обробляється через браузер, який

вбудований в сам додаток. Без наявності підключення до Інтернету гібридний додаток не працює. Останнім часом, більшість компаній вважає за краще віддавати вибір саме гібридному додатку, оскільки даний вид поєднує в собі переваги нативних і веб-додатків [5].

Якщо порівняти з нативними додатками, то гібридне додаток на порядок дорожче, проте швидкість програми вище. Якщо порівнювати з веб-додатком, то і в гібридному можна досить швидко і легко вносити корективи, тобто розробникам немає потреби кожного разу повторно розміщувати додаток в магазині додатків після виправлень помилок в попередній версії. Також, величезним плюсом є той фактор, що розробка гібридного додатки є крос-платформена, і позбавляє необхідності створювати додаток окремо на декількох платформах. Таким чином, підсумовуючи переваги, можна сказати, що гібридний додаток необхідно розробляти, якщо є необхідність заощадити в бюджетному плані, розробити нескладний додаток з простою анімацією, а також, додаток який працює відразу на мінімум 2 платформах. До мінусів можна віднести некоректну роботу при відсутності Інтернету, менш швидку швидкість роботи, в порівнянні з нативними додатками, і мінімалізм у плані візуальних елементів [5].

1.3 Аналіз основних моделей та методів оцінки трудовитрат IT-проекту з розробки мобільних додатків

1.3.1 Аналіз основних інтуїтивних моделей та методів оцінки трудовитрат IT-проекту з розробки мобільних додатків

Використання методологія Agile передбачає на увазі застосування більше інтуїтивних методів оцінки завдань, а також проповідує наступні

принципи до методів оцінки завдань: Перший принцип говорить про те, що в Agile дуже цінується швидкість оцінки завдання, швидкість розробки проекту, оперативний зворотний зв'язок. Сама по собі оцінка не несе ніякої бізнес цінності, тому в динамічному ітеративному процесі логічно зробити її з найменшими затратами праці, як можна дешевше і більш оперативно. Другий принцип полягає в акценті на командну роботу, так як більшість методів є груповими. У методах Agile дуже важливо використовувати думку різних фахівців команди. Даний принцип дозволяє комфортно висловлюватися кожному члену команди. Третій принцип говорить про використання відносних одиниць вимірювання. Даний підхід дозволяє уникати асоціацій і зв'язку до абстрактних величин, наприклад днях або годин [6].

Інтуїтивні методи. Сутність неалгоритмічних методів полягає в тому, що при оцінці вартості ПЗ використовуються певні схеми і принципи, а не математичні формули.

До цих методів належать такі: Price-to-win, оцінка Паркінсона, метод T-Shirt Sizes, метод Planning Poker, метод Bucket System, метод Dot-voting, метод Maximum Size or Less, метод Big/Small/Uncertain, метод Ordering Rule.

Метод Price-to-win ґрунтується на принципі «клієнт завжди правий». Суть методу полягає в тому, що незалежно від передбачуваних реальних витрат на розробку проекту, оцінка вартості ПЗ коригується відповідно до побажань замовника. Price-to-win фактично є результатом узагальнення успішних переговорів з 292 клієнтами, тому часто застосовується компаніями, які не мають коштів для якісної оцінки проектів. Застосування методу може мати для розробника наступні негативні наслідки: брак ресурсів для виконання проекту, невиконання термінів здачі проекту і як результат – втрата контракту або банкрутство [7].

Оцінка Паркінсона . Метод ґрунтується на принципі: «Обсяг роботи зростає в тій мірі, в якій це необхідно, щоб зайняти час, виділений на її

виконання» . Принцип, який назвали «законом» , був вперше висловлений С.Н. Паркінсоном і описував природу взаємодії бюрократичної системи в адміністративних інститутах, відображаючи процес неефективного використання ресурсів. У застосуванні до розробки програмних проєктів, закон Паркінсона використовується у вигляді такої схеми: щоб підвищити продуктивність праці розробника, необхідно зменшити час, відведений на розробку [7].

В Agile часто використовують наступні інтуїтивні методи:

Метод T-Shirt Sizes (розміри футболки). В якості одиниці вимірювання в цій техніці використовується розмір футболки: XS, S, M, L, XL. Команда приймає рішення про розмір тієї чи іншої користувальницької історії в ході спільної відкритої дискусії. У разі невизначеності, можливе застосування голосування. При бажанні можна домовитися про співвідношення «розмірів», наприклад, S це до 2 XS, M це до 2 S і так далі. Як правило, перші кілька завдань оцінюються попередньо. Далі починає вимальовуватися картини щодо інших завдань і вони приймаються за XS. Після цього інші завдання оцінюються з точки зору наскільки вони більше XS. Залежно від цього їм присвоюється ступінь декомпозиції історій. У підсумку ми знаходимо найдрібніший певний розмір S, M, L або XL [6].

Також можна домовитися, що у нас є, наприклад, великий розмір XXL. Присвоєння історії цього розміру говорить, що насправді ми не можемо оцінити завдання і вона потребує подальшої декомпозиції або уточнення. Дана техніка є досить швидкою і її можна використовувати для оцінки великої кількості user story за одну сесію. З її допомогою цілком реально за годину оцінити 15-20 історій [6].

Метод Planning Poker (покерне планування). Це одна з найпопулярніших технік оцінки. Учасники процесу використовують спеціально пронумеровані карти (подібні гральним), щоб голосувати з їх допомогою за оцінку user story. Зазвичай для «покеру» використовуються карти з числами Фібоначчі (0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89), але можливі й

інші варіанти. Процес оцінювання з використанням методу Planning Poker полягає в наступних етапах [6]:

а) кожен учасник отримує колоду карт з числовими значеннями для оцінки, зображенням «?» (запит уточнення завдання) і «чашки кави» (вимога перерви);

б) product owner робить короткий анонс чергової користувальницької історії і відповідає на питання команди по цьому завданню;

в) учасники «покеру» вибирають карту з відповідною на їхню думку оцінкою і кладуть їх сорочкою вгору (щоб не впливати на вибір один одного);

г) після того, як всі члени команди вибрали свої оцінки карти одночасно перевертаються;

д) учасникам з найнижчими і найвищими оцінками роблять короткі коментарі пояснюючи свій вибір оцінки;

е) в результаті процесу обговорення команда приходять до єдиного рішення і після цього переходить до наступної користувальницької історії.

Planning Poker одна з найточніших технік оцінки, але підходить для порівняно невеликої кількості завдань. Протягом годинної сесії таким способом можна оцінити 4-10 історій [6].

Метод Bucket System (система "ведерок"). У цій методиці використовується принцип схожий на Planning Poker – завдання оцінюються і поміщаються з відерця з відповідним розміром. Для вказівки розміру також можна використовувати числа Фібоначчі. Однак у цих методів є принципова відмінність – в Bucket System після початкового масштабування задач, процес завдання розділяються між учасниками для оцінки. Процес оцінювання з використанням методу Bucket System полягає в наступних етапах [6]:

а) всі історії, які потрібно оцінити, виписуються на картки;

б) на столі або дошці вибудовується послідовність з «відерець» для завдань різного розміру;

в) команда вибирає по черзі 3-5 довільні картки з завданнями і оцінює їх в ході відкритого обговорення порівнюючи і вибудовуючи їх один щодо одного;

г) завдання розміщуються у відповідні «відра» задаючи загальний масштаб і орієнтири для подальших оцінок;

д) далі все решта завдання порівну розділяються між усіма учасниками і оцінюються ними самостійно, з урахуванням отриманої шкали вимірювань;

е) якщо хтось із членів команди не може оцінити будь-яку історію, то він передає її іншому.

Даний метод (на відміну від покеру планування) може використовуватися для швидкої оцінки дуже великого числа завдань (від 50) і з великою кількістю учасників [6].

Метод Dot-voting (голосування по точкам). Цей спосіб передбачає використання спеціальних «точок», які показують голоси учасників \ бали оцінки поставлені для того чи іншого завдання. В якості таких «точок» можуть використовуватися: наклейки, стікери, магніти, точки \ штрихи проставлені маркерами. Процес оцінювання з використанням методу Dot-voting полягає в наступних етапах [6]:

а) всі оцінювані User Stories виписуються на окремі картки і розміщуються на столі /дошці;

б) для виконання оцінки кожен з учасників отримує однакову кількість «точок». Кожен член команди розподіляє свої «точки» між завданнями як він вважає за потрібне, враховую, що чим більше «точок», тим складніше завдання і тим більше на неї необхідно часу;

в) після того як кожен учасник зробив свою оцінку і розподілив всі свої «точки», підраховується загальна кількість точок виставлених для кожної користувальницької історії. В результаті всі завдання ранжуються між собою за кількістю «точок».

Даний метод є дуже простим і швидким, він буде ефективно працювати для оцінки невеликої кількості історій (до 8-10) [6].

Метод Maximum Size or Less (поділ до максимального розміру або менше). У рамках цієї техніки учасники процесу оцінки спочатку визначають максимально можливий розмір для завдання у беклогу. Найчастіше в якості максимального значення вибирається 1 людино-день. В цьому випадку найбільші завдання повинні вимагати для їх виконання не більше 1 дня. Процес оцінювання з використанням методу Maximum Size or Less полягає в наступних етапах [6]:

а) кожна історія обговорюється всіма учасниками, щоб відповісти на питання: що оцінюється завдання більше максимального значення або менше;

б) якщо дана історія більше максимального розміру, то група декомпозує її на підзадачі і повторює процес з оцінки для складових частин;

в) процес триває поки все оцінювані завдання не опиняються в дозволеному діапазоні розмірів – дорівнюватимуть або будуть менше обраного за максимальне значення.

Цей метод оцінки також дуже простий у використанні і з його допомогою команда здатна впоратися з 15-30 завданнями (в залежності від складності та досвіду декомпозиції) [6].

Метод Big/Small/Uncertain (великий/малий/невизначений). «Даний метод схожий на техніку Bucket System, основна відмінність полягає в тому, що в цьому випадку використовується тільки 3 відра: великий розмір, малий розмір, невизначений розмір завдання. Процес оцінки з використанням методу Big / Small / Uncertain являє собою послідовність таких етапів [6]:

а) всі оцінювані історії обговорюються учасниками і поміщаються в одну з трьох категорій Big / Small / Uncertain;

б) спочатку група проводить групове обговорення кількох перших завдань (3-5), визначаючи масштаб і орієнтири для кожної категорії;

в) потім, подібно Bucket System, що залишилися історії розподіляються між учасниками і оцінюються самостійно, що сильно прискорює процес.

Це одна з найшвидших технік оцінки. Вона дозволяє оцінити за одну сесію велика кількість історій (від 50) і дозволяє залучати до процесу одночасно багато учасників [6].

Метод Ordering Rule (вибудовування порядку) . Даний метод являє собою покрокову гру, мета якої вибудувати всі завдання один щодо одного на єдиній шкалі розміру. Процес оцінки з використанням методу Ordering Rule являє собою послідовність таких етапів [6]:

- а) спочатку все оцінювані історії виписуються на картки;
- б) картки з завданнями випадковим чином розміщуються на столі або дошці зі шкалою, на кордонах якої вказані «малий розмір» і «великий розмір»;
- в) кожен учасник по черзі робить свій «хід» оцінки. Такий «хід» включає одне з наступних можливих дій: перемістити будь-яку історію за шкалою на одну поділку (тобто поміняти оцінку на більш низьку або високу), обговорити історію з колегами, пропустити свій «хід»;
- г) в результаті «ходів» співробітників завдання можуть переміщатися по дошці, їх оцінка один щодо одного уточнюється;
- д) коли всі учасники пропускають свій «хід», процес оцінки завершується. Всі завдання розподілені за шкалою між значеннями «малий розмір» і «великий розмір».

Цей метод досить ефективний для оцінки невеликої кількості завдань (5-15). Учасники залучені до загального гейміфіційованого процесу і змінюючи положення історій відносно один одного домагаються високої точності оцінки [6].

Загальний недолік інтуїтивних методів полягає в тому, що вони надають доволі приблизний результат оцінювання, так як ґрунтуються на інтуїції, неможливість виключити повністю суб'єктивізм в оцінці експертів.

1.3.2 Аналіз основних формальних моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків

В даний час існує досить велика кількість параметричних моделей і методів оцінювання витрат на виконання ІТ-проекту зі створення ІС, серед яких можна виділити [8]:

- а) спрощений метод функціональних точок;
- б) метод функціональних точок;
- в) метод об'єктних точок;
- г) метод Де-Марко;
- д) метод точок властивостей;
- е) лінійні методи;
- ж) метод Wideband Delphi;
- і) методи, засновані на моделях СОСОМО і СОСОМО II.

Незважаючи на їх різноманітність, більшість даних методів не можна застосовувати в ході ініціації ІТ-проекту з розробки ІС. Так, застосування методів Де-Марко, точок властивостей, лінійних методів, а також методу Wideband Delphi в цьому випадку практично неможливо через неможливість отримання необхідних оцінок з прийнятним рівнем достовірності. У той же час, використання методів спрощених функціональних точок, класичних функціональних точок і об'єктних точок дозволяє отримати наближені, не зовсім точні, але обґрунтовані та придатні для прийняття рішень оцінки [9].

Також одним з найпопулярніших методів оцінки вартості розробки програмного забезпечення стала модель СОСОМО. Дана модель являє собою набір формул з параметрами, котрі характеризують конкретний ІТ-проект. В даний час їй на зміну прийшла вдосконалена модель СОСОМО II, яка більш повно підтримує сучасні процеси розробки ПЗ і більш адекватно оцінює сучасні ІТ-проекти. Варто відзначити, що як метод оцінки витрат СОСОМО

II підходить не для всіх ІТ-проектів. Наприклад, при розробці комерційних додатків, даний метод зазвичай привід до апіорно завищених значень [10].

Таблиця 1.1 – Зведена таблиця недоліків існуючих моделей та методів для оцінки трудовитрат ІТ-проекту з розробки мобільних додатків

Вид моделей/методів	Приклади моделей/методів	Загальні недоліки оцінок
Параметричні оцінки	а) спрощений метод функціональних точок; б) метод функціональних точок; в) метод об'єктних точок; г) метод Де-Марко; д) метод точок властивостей; е) лінійні методи; ж) метод Wideband Delphi; і) методи, засновані на моделях COSOMO і COSOMO II.	– занадто складні для оцінки задач, що пов'язані з розробкою мобільних додатків; – занадто громіздкі (необхідно доволі великий час для розрахунку оцінки); – необхідно збирати доволі велику кількість даних о продукті.
Інтуїтивні оцінки	а) price-to-win; б) оцінка Паркінсона; в) метод T-Shirt Sizes; г) метод Planning Poker; д) метод Bucket System; е) метод Dot-voting; ж) метод Maximum Size or Less; і) метод Big/Small/Uncertain; й) метод Ordering Rule.	– неможливість виключити повністю суб'єктивізм в оцінці експертів; – неможливість забезпечити об'єктивну оцінку компетентності експертів (не дуже точні).

1.4 Аналіз існуючих інформаційних систем моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків

1.4.1 Аналіз існуючих інформаційних систем формальних моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків

У мережі Інтернет було проведено аналіз існуючих інформаційних систем формальних моделей та методів оцінки трудовитрат ІТ-проекту. Аналіз показав, що існує доволі невелика кількість існуючих ІС, які можуть виконувати функцію оцінки трудовитрат ІТ-проекту. Велика частка – це платні ІС. Проте, навіть вони не спеціалізовані саме для оцінки трудовитрат ІТ-проекту з розробки мобільних додатків.

Сервіс «SoftwareCost». Був проведений пошук веб-ресурсу або додатки, який би реалізував метод СОСОМО II. Таким чином, був знайдений сервіс «SoftwareCost» Цей сервіс є калькулятор, що дозволяє автоматизувати розрахунки таких величин як очікувані трудовитрати на створення ІТ-проекту, передбачуваний час виконання ІТ-проекту, потреба в персоналі та продуктивність персоналу ІТ-проекту [7].

Екранна форма даного сервісу, що дозволяє задавати вихідні значення для розрахунків за моделлю СОСОМО II, показана на рисунку. 1.2. Екранна форма даного сервісу, що дозволяє відобразити результати розрахунків, показана на рисунку. 1.3.

Рисунок 1.2 – Екранна форма калькулятора моделі СОСОМО ІІ, що дозволяє задавати вихідні значення для розрахунків

Results

Software Development (Elaboration and Construction) Staffing Profile

Effort = 0.0 Person-months
Schedule = 0.0 Months
Cost = \$0

Total Equivalent Size = 0 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.0	0.0	0.0	\$0
Elaboration	0.0	0.0	0.0	\$0
Construction	0.0	0.0	0.0	\$0
Transition	0.0	0.0	0.0	\$0

Software Effort Distribution for RUP/MBASE (Person Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.0	0.0	0.0
Environment/CI	0.0	0.0	0.0	0.0
Requirements	0.0	0.0	0.0	0.0
Design	0.0	0.0	0.0	0.0
Implementation	0.0	0.0	0.0	0.0
Assessment	0.0	0.0	0.0	0.0
Deployment	0.0	0.0	0.0	0.0

Your output file is http://casie.usc.edu/tools/data/COCOMO_june_17_2014_23_44_33_200986.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rmadach@nps.edu

Рисунок 1.3 – Екранна форма калькулятора моделі СОСОМО ІІ, що дозволяє відобразити результати розрахунків

Costar. Засоби, засновані на моделі СОСОМО. Допускається використання всіх реалізацій моделі СОСОМО, моделей життєвого циклу ПО Waterfall і MBASE / RUP, підтримується робота з проектом, складеним з компонентів, для кожного з яких можна виконати роздільну оцінку. Costar дозволяє проводити оцінку в двох режимах: покроковому, за допомогою майстра оцінки вартості; інтерактивному, що забезпечує безпосередню вказівку значень параметрів, що впливають на вартість проекту. Для визначення розміру оцінюваного проекту використовуються функціональні точки або рядки коду. Для перекладу значень, зазначених в рядках коду, в

програмі є конвертер, що розраховує значення розміру коду в функціональних точках виходячи з мови програмування, який використовується для реалізації проекту. Costar підтримує обмеження проекту, засновані на граничних фінансових витратах і крайній термін реалізації проекту [7].

Для оцінки витрат, пов'язаних з оплатою праці працівників існує два альтернативних підходи: розрахунок витрат для кожного з етапів життєвого циклу ПЗ; розрахунок місячної оплати праці для кожної категорії співробітників. Для аналізу результатів оцінки Costar створює різні форми звітів, графіків і діаграм. Звіти, представлені у формі таблиць, можуть бути збережені у форматі Microsoft Excel, графіки та діаграми – в форматі растрового зображення BMP [7].

Для проведення точної оцінки вартості розробки ПЗ модель СОСОМО вимагає детального і різнобічного опису проекту. Це може ускладнити застосування заснованих на ній коштів на ранньому етапі розробки ПО, і сприяє підвищенню точності оцінки на пізніх етапах розробки ПЗ, при аналізі завершеного проекту. При використанні коштів на основі моделі СОСОМО або СОСОМО II факторами, що впливають на точність оцінки вартості є такі: правильний вибір конкретної реалізації моделі СОСОМО; точність калібрування – відповідність установок вихідними даними. У зв'язку з цим, для застосування засобів використовують персонал, який не має прямого відношення до процесів проектування і розробки ПЗ. Він формує специфікації проекту і параметри, необхідні для оцінки, які надаються співробітникам, які виконують оцінку [7].

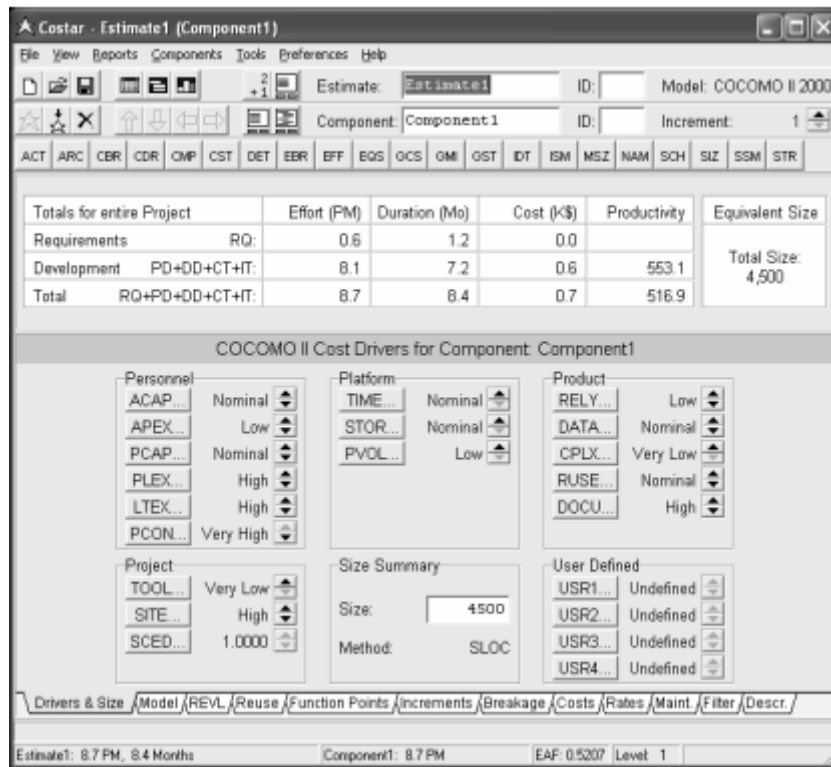


Рисунок 1.3 – Екранна форма калькулятора Costar моделі COCOMO II, що дозволяє задавати вихідні значення для розрахунків

1.4.2 Аналіз існуючих інформаційних систем інтуїтивних моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків

Pointing Poker. Переваги даного сайту [11]:

- безкоштовний і існує за рахунок донатів;
- не вимагає реєстрації, можна просто створити нову сесію або приєднатися до існуючої і почати голосувати;
- ви можете встановити шкалу під себе;
- можна вибрати в налаштуваннях, хто може відкривати / скидати карти.

Недоліки даного сайту:

– учасники можуть змінювати свої голоси навіть після того, як карти відкриті;

– не дозволяє експортувати список сторей для оцінки, можна тільки копіювати опис сторі перед кожним туром голосування – це незручно і займає час;

– неможливо зберігати історію оцінок попередніх сесій, щоб повернутися до неї пізніше при необхідності (в нашому випадку це дійсно корисно, оскільки ми переносимо всі оцінки в JIRA, і іноді хочеться перевірити якусь із оцінок) [11].

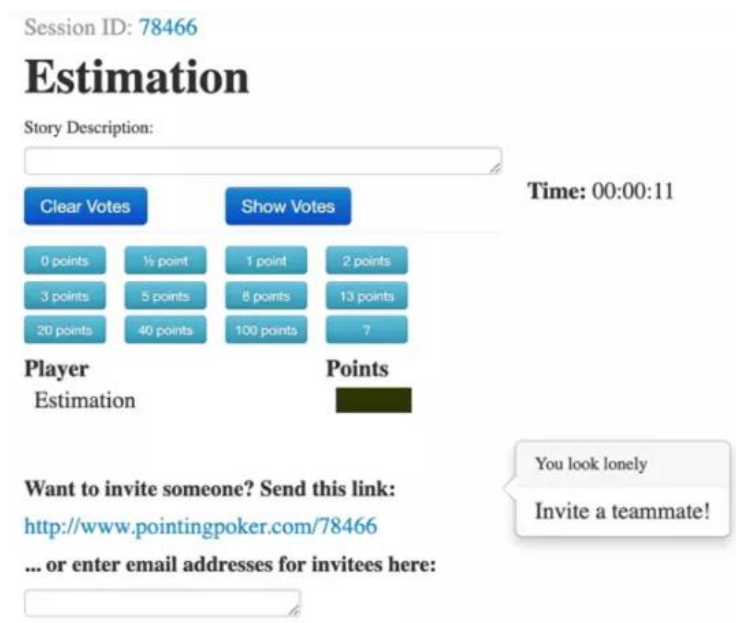


Рисунок 1.4 – Екранна форма калькулятора Estimation

Scrum poker Online . Переваги даного сайту [11]:

– безкоштовний;
 – є набір різних видів шкал на випадок, якщо ви використовуєте не Фібоначчі, а «футболки» або щось ще.

Недоліки даного сайту [11]:

– тільки творець сесії може бачити повну картину, інші учасники бачать тільки свої голоси навіть після відкриття карт. Розробники даного

сервісу рекомендують «відображати вид Scrum Master на великому екрані (телевізорі або проекторі) і дозволити всім іншим приєднатися через смартфон», але нам це було зовсім не зручно;

- власник сесії не може голосувати: нас це влаштовувало, але є команди, яким це може бути не зручно;

- немає можливості зберігати історію сесій і не можна заздалегідь скопіювати і вставити список історій, як це було з Pointing Poker.

Заявлена інтеграція з JIRA / GitLab / GitHub, але ми ніколи не використовували її, так як не хотіли відкривати доступ до нашої JIRA для будь-яких зовнішніх інструментів, в яких ми не впевнені на 100% [11].

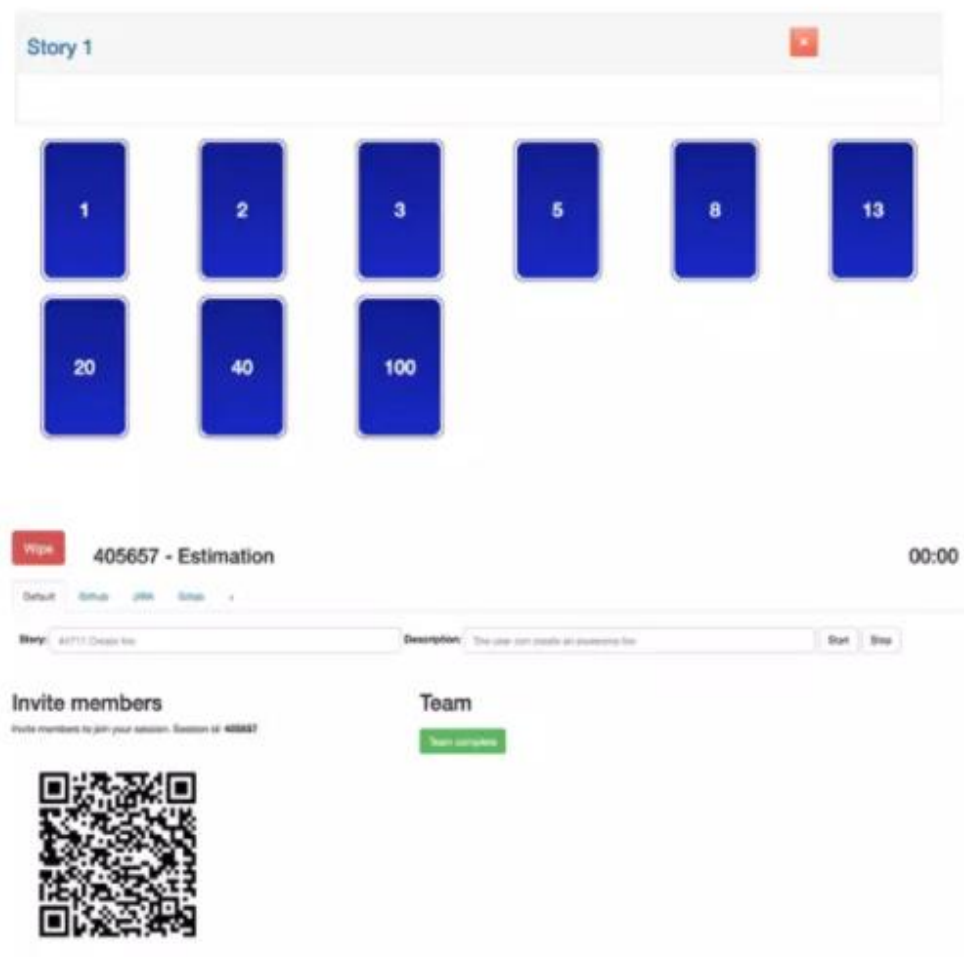


Рисунок 1.5 – Екранна форма калькулятора ScrumPoker Online

1.5 Формування мети та постановка задачі кваліфікаційної роботи

Існуючі моделі оцінки трудовитрат ІТ-проекту з розробки мобільного застосування, наприклад, ті ж СОСОМО 1 і СОСОМО 2, можуть не підійти для об'єктивної оцінки трудовитрат по розробці мобільного додатку, дані методи недостатньо вивчені для використання оцінки таких видів додатків.

Часто оцінка трудовитрат ІТ-проекту по розробці мобільного додатку проводиться суб'єктивно, :суб'єктивність оцінки, залежність оцінки проекту від рівня навичок ІТ-фахівця. Також, існує доволі велика кількість agile-методів для оцінки задач, які мають високу швидкість оцінки. Авжеж, у певних випадках вони можуть бути доволі ефективними, проте , вони мають доволі великий недолік – низький рівень точності оцінки завдання.

Метою досліджень кваліфікаційної роботи є дослідження моделей та методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків, та розробка або модифікація моделі або методу оцінки даної задачі, який буде більш ефективний, ніж існуючі моделі або методи та легкий у використанні.

Для досягнення мети в роботі пропонується вирішити такі задачі дослідження:

- аналіз існуючих методів оцінки трудовитрат з розробки мобільних додатків;
- розробка модифікованого методу «секторів»;
- опис практичного вирішення задачі оцінювання трудовитрат розробки ІТ-проекту з розробки мобільного додатку;
- апробація методу секторів під час оцінювання трудовитрат ІТ-проекту з розробки мобільного додатку.

2 РОЗРОБКА МОДИФІКОВАНОГО МЕТОДУ «СЕКТОРІВ»

2.1 Синтез модифікованого методу «Секторів» за результатами інтеграції методів «Big/small/Uncertain», «Maximum Size or Less» та «Ordering Rule»

В першій частині кваліфікаційної роботи було проведено аналіз існуючих інтуїтивних та формальних моделей та методів, й було отримано наступні результати:

- описано принципи роботи розглянутих моделей та методів;
- описано ситуації, коли їх краще використовувати.

На підставі аналізу існуючих недоліків інтуїтивних методів, що описані в 1-му розділі, було вирішено створити новий модифікований метод оцінювання трудовитрат ІТ-проектів з розробки мобільних додатків.

Ідея створення модифікованого методу представлена в тезисах XXV Міжнародного Молодіжного Форуму м.Харків [12].

Новий модифікований метод, названо методом «Секторів», об'єднує у собі деякі принципи роботи інтуїтивних методів, а саме: «Big/small/Uncertain», «Maximum Size or Less» та «Ordering Rule».

Сутністю модифікованого методу «Секторів» є інтеграція з виправленням недоліків методів «Big/small/Uncertain», «Maximum Size or Less» та «Ordering Rule» та використанням чисел Фібоначчі для підвищення точності оцінки трудовитрат User Stories.

Також, для конкретизації точності оцінювання User stories (великих, середніх та малих) задач вирішено використати ітеративний підхід.

Розроблений метод дозволить доволі точно визначити кількість трудовитрат для виконання необхідного списку User stories, є доволі швидким у використанні та простим для розуміння.

Для прикладу, його можна буде використовувати для планової оцінки великої кількості задач (50-70) на наступний місяць.

2.2 Етапи оцінювання вартості ІТ-проекту з використанням модифікованого методу

Етап 1. Розподілення всією Scrum Team разом перших 3-6 User Stories між беклогів кругів S, M та L. У разі дуже великих чи незрозумілих User Stories відкладення їх до беклогу для декомпозиції.

Етап 2. Індивідуальне та паралельне розподілення членами Scrum Team User Stories, що залишились у вхідному беклогу між беклогами кругів S, M та L. У разі дуже великих чи незрозумілих User Stories відкладення їх до беклогу для декомпозиції.

Етап 3. Розподілення членами Scrum Team User Stories між секторами круга S, використовуючи певний набір правил та дій.

Етап 4. Переміщення User Stories з сектору N кола S до беклогу кола M.

Етап 5. Розподілення членами Scrum Team User Stories між секторами круга M, використовуючи певний набір правил та дій.

Етап 6. Переміщення User Stories з сектору N кола M до беклогу кола L.

Етап 7. Розподілення членами Scrum Team User Stories між секторами круга L, використовуючи певний набір правил та дій.

Етап 8. Переміщення User Stories з сектору D кола L до беклогу для декомпозиції.

Етап 9. Декомпозивання усіх User Stories, що знаходяться у беклогу для декомпозивання.

Етап 10. Повторення послідовності дій, починаючи з етапу 2.

Етап 11. Розміщення оцінених User Stories у вихідному беклогу (верхня частина дошки) у порядку зростання складності.

Етап 12. У разі необхідності уточнення під час обговорення певних User Stories членами Scrum Team та зміна кількості Story Points у них.

Інформація про використання етапів та описом модифікацій похідних методів у модифікованому методі «Секторів» наведена у таблиці 2.1.

Таблиця 2.1 – Зведена таблиця з використанням етапів та описом модифікацій похідних методів у модифікованому методі «Секторів»

№	Похідний метод та його етап	Суть модифікації похідного методу	Етап методу секторів, в якому використовується етап похідного методу
1	2	3	4
1	Метод «Big/small/Uncertain». Етап 1.	Запропоновано розподіляти перші 3-6 User Stories між колами замість ведер	Етап 1
2	Метод «Big/small/Uncertain». Етап 2.	Запропоновано розподіляти індивідуально та паралельно членами Scrum Team User Stories, що залишились між колами замість ведер	Етап 2
3	Метод «Maximum Size or Less». Весь метод.	Замість 1 людино-дня для визначення трудовитрат будуть використовуватись числа Фібоначчі, максимальне буде 34 Story Points. Якщо трудовитрати будуть більше, то User Story повинна бути декомпозована.	Етап 1 та Етап 2

Кінець таблиці 2.1

1	2	3	4
4	Метод «Ordering Rule». Весь метод.	1. Поява чисел Фібоначчі, що дозволяє доволі точно оцінити трудовитрати User Stories. 2. Використання замість прямої шкали, 3 круги (S,M,L), поділених на сектори	Етапи 3-9

Наприклад, існує деяка кількість User Stories – N, треба оцінити їх трудовитрат (в Story Points). Для цього буде використано «Метод секторів».

Для визначення трудовитрат User Stories буде брати участь Scrum Team. Зазвичай, кількість членів команди, відповідно до Scrum Guide, дорівнює від 6 до 10 людей.

Етап 1. Члени Scrum Team разом оцінюють перші 3-6 User Stories. Це необхідно для того, щоб визначити масштаб й орієнтири для подальшої оцінки User Stories, щоб всі розуміли на певних прикладах, які види User Stories маються на увазі під простим, середнім, та великим рівнем трудовитрат.

Треба зазначити, що необхідно брати таку кількість User Stories при розподіленні усією Scrum Team, щоб хоча б одна User Story була у кожному кругу.

У разі дуже великих чи незрозумілих User Stories відкладення їх до беклогу для декомпозиції.

Етап 2 . Тепер, User Stories що залишились, розподіляються рівномірно між усіма членами Scrum Team й кожен член команди повинен розподілити великими «мазками» ці User Stories, що він має, між трьома беклогами кругів S, M, та L. Якщо він має сумніви, то він може проконсультуватись у того члена команди, котрого він вважає більш компетентним у питанні, що пов'язане зі специфікою певного User stories. Якщо ж він взагалі не розуміє ,

як оцінити, навіть орієнтовно певне User story, то він може віддати його іншому члену Scrum Team, котрого він вважає більш компетентним у цьому питанні.

При оцінці, можуть виникнути проблеми щодо оцінювання певних User stories. В такому випадку, їх необхідно буде декомпонувати, або уточнити у замовника додатково інформацію, щоб зрозуміти, що конкретно він мав на увазі. Незрозумілі User stories Scrum Team розташовує у верхньому беклогу (верхня частина дошки).

Етап 3-8. Далі, необхідно поетапно розподілити User stories в кожному з секторів кола S, M, L.

Коло S (small) – це коло, до котрого потрапляють User Stories з маленькою кількістю трудовитрат.

Коло M (middle) – це коло, до котрого потрапляють User Stories з середньою кількістю трудовитрат.

Коло L (large) – це коло, до котрого потрапляють User Stories з великою кількістю трудовитрат.

Кожне з колів розділено на сектори. Кожен з секторів має своє значення або позначку.

Для оцінки User stories було використано числа Фібоначчі, а також цифру $\frac{1}{2}$ для додаткової зручності. Числа Фібоначчі використовуються в даному методі, оскільки вони мають цікаву особливість – сума двох попередніх дорівнює наступному числу. Таким чином, команда може оцінити, які завдання краще буде взяти у реалізацію – більш складну User story, яке, наприклад, має складність 8, або двоє менших – 5 та 3. Можливо, дві менших User stories дадуть більшу бізнес-цінність клієнту на даному етапі розробки програмного забезпечення.

Позначка N у секторі означає перекладання User Story у інше коло, оскільки може виявитись ситуація, що після обговорення членами Scrum Team, при оцінці великими мазками, один з членів команди помилково розподілив певне User story в круг не відповідної складності.

Позначка D у секторі означає, що певне User Story необхідно буде декомпонувати, оскільки після його поетапного оцінювання виявилось, що воно занадто складне, або незрозуміле для того, щоб було оцінено.

Коло з центром S – це коло з невеликою складністю User Stories (від ½ до 2), M – це коло з середньою складністю User Stories (від 3 до 8), та L – це коло з великою складністю User Stories (13, 21, 34 й більше).

Тепер, кожен член команди буде покрово оцінювати User Stories у колах S, M та L.

Спочатку, члени Scrum Team починають оцінювати User Stories з S, далі переходять у коло M, а далі у коло L.

Алгоритм дій членів Scrum Team виглядає наступним чином:

Кожен член команди Scrum Team в кожному колі кожної ітерації має три ходи. У кожному з ходів член Scrum Team має три можливих дії:

- взяти User Story з беклогу кола, з яким зараз працює Scrum Team та оцінити його, поклавши у певний сектор кола. Якщо ж, член Scrum Team сумнівається, він може попросити поради у того члена команди, котрого він вважає більш досвідченим та компетентним для оцінки певного User Story, після отримання поради, він оцінює дане User Story;

- перекласти User Story у сусідній сектор кола з яким працює. Сусідній сектор – це той сектор, котрий знаходиться лівіше, або правіше від того, на котрому знаходиться зараз певне User Story. Одразу перекладати певне User Story на декілька рівнів заборонено. Це необхідно для того, щоб якийсь з членів Scrum Team не давив свої авторитетом. Треба зазначити, що на першому ході першого члена Scrum Team на дошці ще немає User Story, які можна було б перекладати;

- пропустити хід.

Таким чином, даний послідовність дій повторюється до тих пір, поки кожен з членів команди не почне пропускати свій хід. Це означає, що всі члени команди згодні з тим, яким чином розташовані User Stories у секторах певного кола.

Кількість ітерацій розподілення User Stories у секторах колах залежить від того, скільки User Stories взагалі є у певному колі (після розподілення «великими мазками»), та від того, коли всі члени Scrum Team будуть задоволені розподіленням User Stories у секторах певного кола та почнуть всі пропускати хід.

Етап 9. Далі декомпозуються усі User Stories, що знаходяться у беклогу для декомпозиції.

Етап 10. Послідовність дій повторюється спочатку з моменту розподілення великими «мазками» User Stories між трьома беклогами кругів S, M, та L.

Етап 11. Scrum Team розміщує оцінені User Stories у вихідному беклогу (верхня частина дошки) у порядку зростання складності.

Етап 12. Також, треба зазначити, що на даному етапі Scrum Team, продивившись усі оцінені User Stories у фінальному вигляді, може провести загальну дискусію, якщо у деяких членів Scrum Team виникли питання, та якщо необхідно, члени Scrum Team можуть змінити розташування певних User Stories, якщо після обговорення, виникла така необхідність.

Таким чином, після використання методу «секторів», Scrum Team доволі швидко та точно оцінили доволі велику кількість User Stories.

2.3 Модифікація методів «Big/small/Uncertain», «Maximum Size or Less»

2.3.1 Використання методу «Big/small/Uncertain» для модифікації

Даний метод пропонується використовувати для опису першого та другого етапу розроблюваного модифікованого методу «Секторів». Стандартний варіант методу «Big/small/Uncertain» дозволяв розподілити членам команди великими «мазками» завдання, що треба було оцінити між

трьох ведер – для маленьких завдань, великих завдань та завдань, складність котрих доволі проблематично оцінити.

У методі «Секторів» метод «Big/small/Uncertain» було модифіковано.

Замість ведер тепер буде використовуватись три кола, котрі поділені на сектори. Кожне коло матиме свій беклог й призначено для задач трьох розмірів – великих (коло L), середніх (коло M), та маленьких (коло S).

Додавання кола для задач середніх розмірів дозволяє більш деталізовано оцінювати їх Scrum Team.

Етап 1. Як і у методі «Big/small/Uncertain» перші 3-6 задач розподілялись разом одразу усією командою, це необхідно для того, щоб визначити масштаб та орієнтири для кожної з категорій.

Етап 2. Інші задачі будуть розподілятися між усіма членами команди й далі вони будуть оцінювати свої User Story самостійно паралельно. Кожен член команди буде мати однакову кількість задач. Кількість задач буде розподілена прямопропорційно відносно кількості всіх членів команди.

Якщо у членів команди виникнуть питання щодо оцінки, вони можуть проконсультуватись у більш досвідчених колег, якщо навіть після поради незрозуміло куди розподілити задачу, то дане User Story може буде передано іншому члену команди.

Загалом, метод «Big/small/Uncertain» є одним з найбільш швидких технік оцінки. Він дозволяє за одну сесію оцінити більше 50 User Stories та дозволяє залучати велику кількість членів команди.

2.3.2 Використання методу «Maximum Size or Less» для модифікації

У методі «Секторів», як і у методі «Maximum Size or Less» є можливість декомпонувати завдання. Даний метод пропонується використовувати також

першому та другому етапу розроблюваного модифікованого методу «Секторів».

Дана можливість буде у членів команди під час етапу розподілення великими «мазками» між беклогами трьох кіл – S, M, L.

Проте, у порівнянні з методом «Maximum Size or Less» у методі «Секторів» буде максимальний фіксований розмір складності для User Story, при перевищуванні котрого, воно повинно бути декомпозовано. Дана складність не повинна перевищувати 34 Story Points.

Ця цифра була обрана тому, що зазвичай в ІТ-компаніях задачі розміром більше не допускають для розробки, команда їх спочатку декомпозує.

Також, саме завдання може бути переміщено до декомпозивання, якщо воно незрозуміле для членів команди та потребує уточнення у замовника додаткових деталей.

Метод «Maximum Size or Less» доволі простий у використанні, за його допомогою команда могла деком позувати 15-30 User Stories за годину (в залежності від складності та досвіду декомпозивання).

2.3.3 Використання методу «Ordering Rule» для модифікації

Метод «Ordering Rule» пропонується використовувати з третього по дев'ятий етап розроблюваного модифікованого методу «Секторів».

Замість позначок «Small» та «Big» тепер використовуються 3 круги, що поділені на сектори. Кожен сектор має цифри Фібоначчі (від $\frac{1}{2}$ до 34, або позначку N або D).

Механізм розподілення задачі у середині сектору схожий з тим, що використовується у методі «Ordering Rule».

Проте, у модифікованому методі «Секторів» члени команди тепер не просто показують, що одна задача більш складна або менш складна ніж інша,

а конкретизують складність задачі за допомогою чисел Фібоначчі, що знаходяться у секторах.

Як і у методі «Ordering Rule» члени команди мають три види ходів: розподілити певну задачу на певний сектор, проконсультуватись у більш досвідченого гравця й тоді розподілити задачу або пропустити хід.

Таким чином розподілення відбувається у всіх секторах.

Колі всі задачі розподілені, то послідовність дій ітераційно повторюється для декомпозованих задач.

У кінці, задачі сортуються у порядку зростання складності у вихідному беклогу.

Цей метод добре зарекомендував себе для оцінки невеликої кількості задач (5-15). Використовуючи можливість зміни однієї задачі відносно іншої, члени команди добавиться доволі високої точності при оцінюванні.

Таким чином, можна зробити наступні висновки. У підрозділі 2.1 було коротко описано загальні принципи модифікованого методу «секторів», результатом котрого є інтеграція з виправленням недоліків методів «Big/small/Uncertain» , «Maximum Size or Less» та «Ordering Rule» та використанням чисел Фібоначчі для підвищення точності оцінки трудовитрат User Stories. Також описано, коли та для якої кількості буде доцільно використовувати модифікований метод.

У підрозділі 2.2 коротко описано етапи модифікованого методу «секторів», приведена зведена таблиця з використанням етапів та описом модифікацій похідних методів у модифікованому методі «Секторів» та ретельний опис кожного з етапів методу «секторів».

У підрозділі 2.3 ретельно описана модифікація кожного з методів, а саме : методу «Big/small/Uncertain», Maximum Size or Less» та «Ordering Rule». Прописано причина використання тих чи інших підходів для оцінки трудовитрат ІТ-проекту, а також, прописана ефективність кожного з методів оцінки трудовитрат ІТ-проекту без модифікації.

3 ОПИС ПРАКТИЧНОГО ВИРІШЕННЯ ЗАДАЧІ ОЦІНЮВАННЯ ТРУДОВИТРАТ РОЗРОБКИ ІТ-ПРОЕКТУ З РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ

3.1 Розробка алгоритму реалізації модифікованого методу оцінювання трудовитрат ІТ-проекту з розробки програмного додатка

Схема роботи методу «Секторів» зображено на рисунку 3.1 [13].

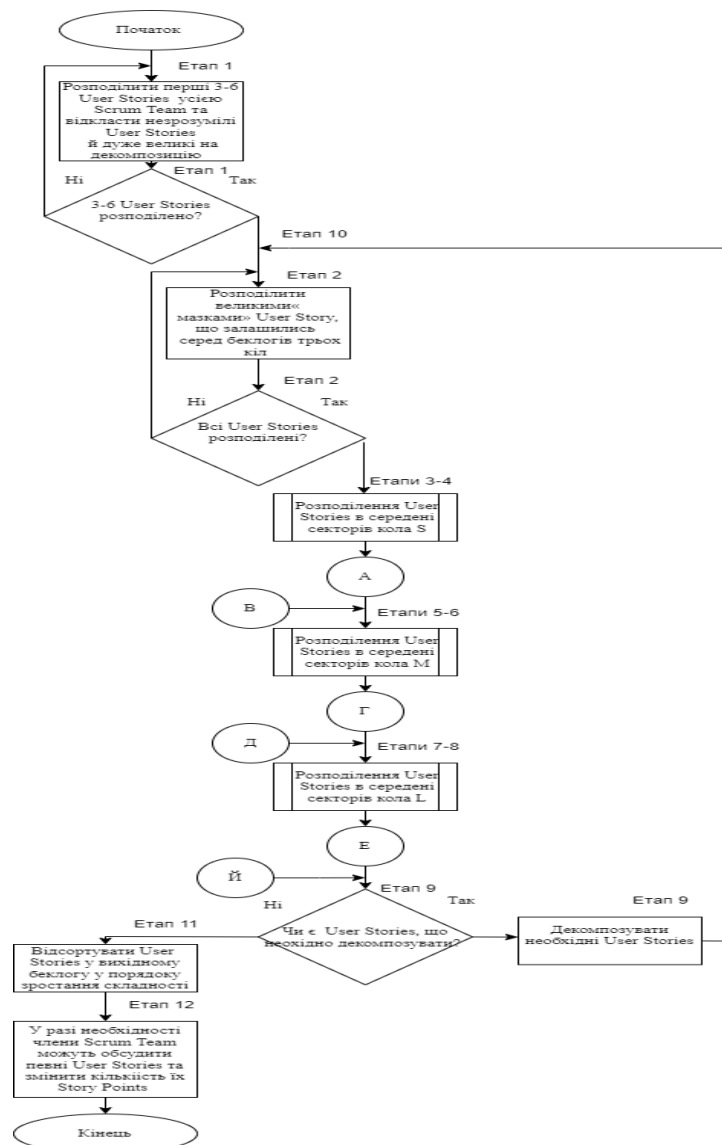


Рисунок 3.1 – Схема роботи методу «Секторів», аркуш 1

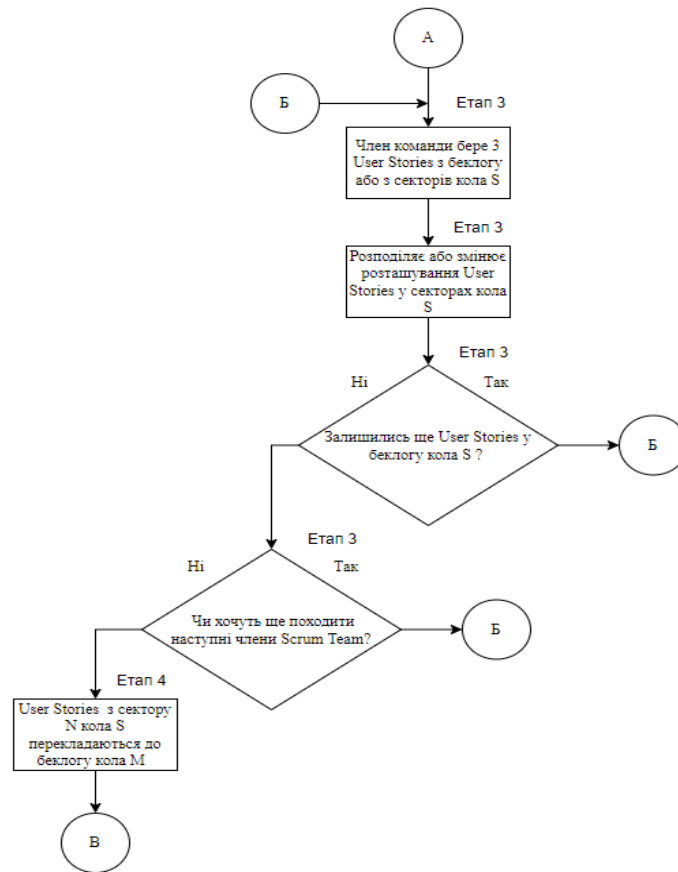


Рисунок 3.1, аркуш 2

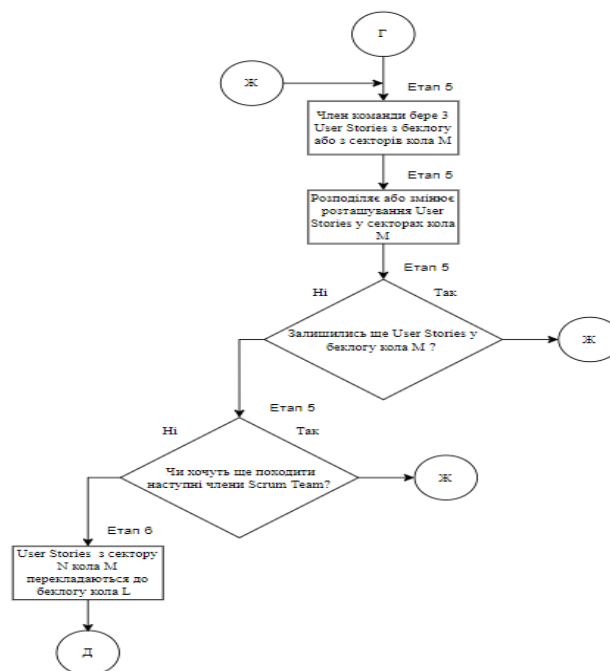


Рисунок 3.1, аркуш 3

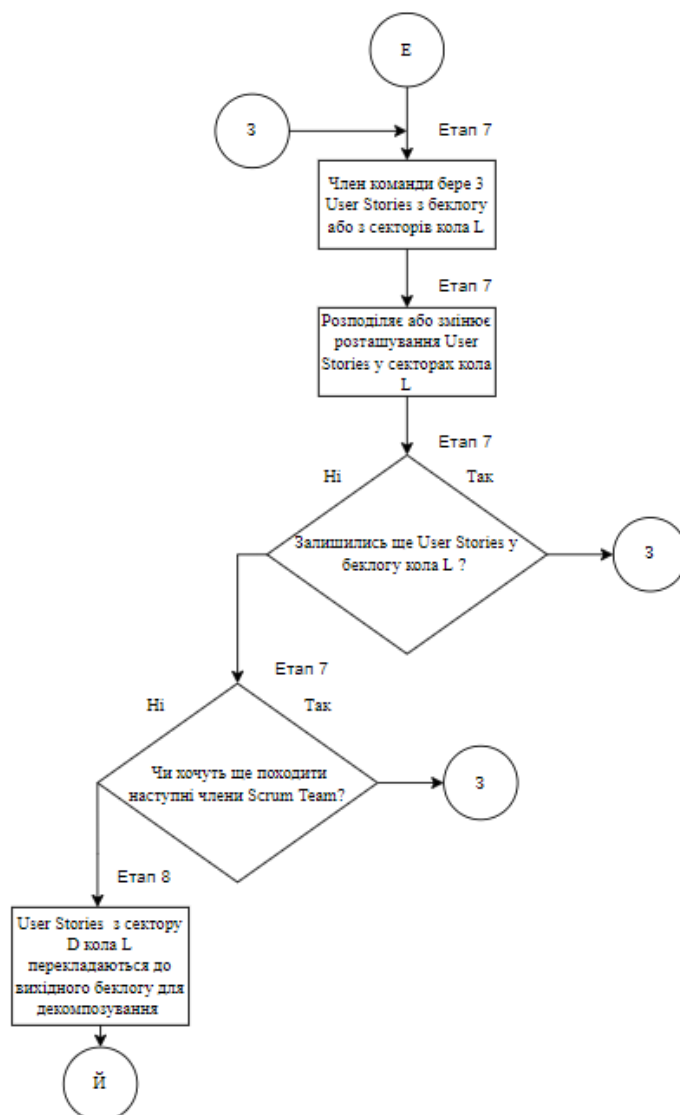


Рисунок 3.1, аркуш 4

3.2 Розробка інструкції членам Scrum Team з виконанням розробленого методу

Етап 1. Scrum Team бере 3-6 User Stories та, обговорюючи разом, розподіляє їх між 3-х кругів.

Якщо якась User Story незрозуміла, то члени команди відкладають її у верхній беклог для подальшої декомпозиції.

Етап 2. Далі, кожен член Scrum Team бере рівну кількість User Stories. Якщо кількість виходить не однакова для членів Scrum Team, то вони розподіляються за остаточним принципом.

Члени Scrum Team самостійно й одночасно розподіляють User Stories між 3-кругів (S, M, та L). Якщо є запитання, то член Scrum Team може проконсультуватись у більш компетентного колеги, або віддати User Story іншому члену команди. Якщо член Scrum Team розуміє, що User Story дуже велике, то він відкладає його до вихідного беклогу для подальшої декомпозиції.

Етап 3. User Stories розподіляються послідовно між 3-ма кругами – спочатку у секторах круга S, потім у секторах круга M, а потім у секторах круга L за принципом, що приведено нижче.

Кожен член команди Scrum Team (при обговоренні в кожному колі) для кожної ітерації має три ходи. Ходи члени Scrum Team виконують послідовно. У кожному з ходів член Scrum Team має три можливих дії:

- взяти User Story з беклогу кола, з яким зараз працює Scrum Team та оцінити його, поклавши у певний сектор кола. Якщо ж, член Scrum Team сумнівається, він може попросити поради у того члена команди, котрого він вважає більш досвідченим та компетентним для оцінки певного User Story, після отримання поради, він оцінює дане User Story;

- перекласти User Story у сусідній сектор кола з яким працює. Сусідній сектор – це той сектор, котрий знаходиться лівіше, або правіше від

того, на котрому знаходиться зараз певне User Story. Одразу перекладати певне User Story на декілька рівнів заборонено. Це необхідно для того, щоб якийсь з членів Scrum Team не давив свої авторитетом. Треба зазначити, що на першому ході першого члена Scrum Team на дошці ще немає User Story, які можна було б перекладати. Також, треба мати на увазі, якщо член Scrum Team захотів перемістити у інший сектор вже розподілену User Story, то він обов'язково повинен прокоментувати причину, чому так він вирішив зробити. В даній ситуації може зав'язатись дискусія, що може допомогти члену Scrum Team знайти найбільш оптимальний варіант розташування певної User Story;

– пропустити хід.

Кількість ітерацій повторюється до тих пір, коли всі члени команди почнуть пропускати ходи. Треба зазначити, що навіть коли всі User Stories взяті з беклогу певного кола, гравці можуть змінювати розташування User Stories під час своїх ходів, проте, треба обов'язково прокоментувати причину.

Етап 4. Якщо після розподілу User Stories серед секторів кола S деяка кількість з них потрапила до сектору N , то вони перекладаються до беклогу кола M.

Етап 5. Аналогічні дії з етапу 3 й повторюються для сектору M.

Етап 6. Якщо після розподілу User Stories серед секторів кола M деяка кількість з них потрапила до сектору N , то вони перекладаються до беклогу кола L.

Етап 7. Аналогічні дії з етапу 3 й повторюються для сектору L.

Етап 8. Якщо після розподілу User Stories серед секторів кола L деяка кількість з них потрапила до сектору D , то вони перекладаються до беклогу для декомпозиції.

Етап 9. Після розподілення усіх User Stories, декомпозиуються ті User Stories, що були відкладені на 1, 2 етапах та 8 для подальшого декомпозиування.

Етап 10. Далі вся послідовність дій починається заново з етапу 2, пропускаючи 1 етап, оскільки всі члени команди вже орієнтуються в принципах оцінювання User Stories.

Етап 11. Після того, як всі User Stories оцінено, то Scrum Team розміщає всі їх у вихідному беклогу (верхня частина дошки) у порядку зростання складності.

Етап 12. Якщо є необхідність, то Scrum Team може обговорити певні User Stories після фінального оцінювання та змінити кількість Story Points.

4 АПРОБАЦІЯ МЕТОДУ СЕКТОРІВ ПІД ЧАС ОЦІНЮВАННЯ ТРУДОВИТРАТ ІТ-ПРОЕКТУ З РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ

Продуктом, який розробляється в рамках проекту «Гід по спеціальностям закладу вищої освіти» є мобільний додаток для операційної системи (ОС) Android. Він складається з двох основних складових: програмного забезпечення (ПЗ); бази даних (БД).

Прикладом ЗВО, для якого розробляється проект, є Харківський національний університет радіоелектроніки.

Даний додаток дозволяє орієнтувати за професією абітурієнтів, переглядати інформацію про спеціальності й кафедри, що існують в ЗВО, легко (централізовано) знаходити абітурієнтам здобувачів вищої освіти певного напрямку, щоб задати питання, що їх цікавлять; надає можливість залишати здобувачам вищої освіти відгуки про навчання за певним напрямом й певної кафедри, а також можливість писати поради абітурієнтам щодо вступу, навчання й студентського життя.

Також, у цьому додатку здобувач вищої освіти певного напрямку може написати, ким він робить після закінчення ЗВО, й які навички з періоду навчання йому знадобились на місці його фахової діяльності.

Серед переваг розроблюваного ПЗ можна виділити новизну підходу (централізованого) щодо пошуку здобувачів вищої освіти для спілкування певної освітньої програми ЗВО, надання дійсно релевантної інформації щодо навчання за певною освітньою програмою на кафедрі, надання релевантної інформації щодо опису тих навичок за певною спеціальністю, що дійсно знадобились на роботі.

У додатку також можна знайти інформацію про те, ким може працювати певний здобувач вищої освіти після закінчення навчання за певною освітньою програмою.

Планується, що мобільний додаток буде безкоштовний й без реклами. Це робиться для того, щоб привернути увагу великої кількості цільової аудиторії.

Сферою використання проекту є сфера освіти.

На даний момент існує проблема щодо профорієнтації абітурієнтів й зручного та централізованого способу спілкування зі здобувачами вищої освіти певної спеціальності.

Основна цільова аудиторія користувачів – це абітурієнти та здобувачі вищої освіти закладу вищої освіти на прикладі Харківського національного університету радіоелектроніки (ХНУРЕ).

Також, користувачами можуть бути й інші зацікавлені особи, такі як викладачі, представники бізнесу тощо.

В подальшому даний мобільний додаток може бути масштабований й на інші ЗВО України.

Проте, однією з головних цілей розробки даного мобільного додатку залишається створення користі абітурієнтам, що не можуть визначитись з майбутньою спеціальністю.

Результатом виконання проекту є програмний продукт (ПП) у вигляді зручного мобільного додатку для Android, що являє собою комплексне рішення, яке складатиметься із двох основних частин: серверна частина та клієнтська частина.

Сервер відповідатиме за збір, обробку та видачу інформації розробнику.

ПП є інтерфейсом, завдяки якому відбуватиметься взаємодія користувача із додатком.

Через це ПП має надавати наступні функції:

- легкий та зручний перегляд каталогу описів спеціальностей ЗВО;
- централізований пошук здобувачів вищої освіти певної спеціальності та кафедри для спілкування;

- введення здобувачами вищої освіти відгуків та порад щодо навчання на певній кафедрі й спеціальності;

- введення випускниками ЗВО /працевлаштованих здобувачів вищої освіти опису необхідних навичок й знань певної спеціальності, що дійсно знадобились на роботі з періоду навчання у ЗВО.

ПП слугує як отримувачем так і відправником інформації про користувача. Для отримання інформації необхідне підключення до мережі Інтернет.

Дерево цілей – наочне графічне зображення підпорядкованості та взаємозв'язку цілей, що демонструє розподіл загальної мети або місії на підцілі, завдання та окремі дії. Основним підходом при побудові дерева є декомпозиція. Сутність методу полягає у розкритті структури системи та досяганні генеральної цілі через виконання менших цілей, що знаходяться на нижчих рівнях.

Також при побудові дерева варто враховувати повноту редукції. Цілі нижчого рівня повинні бути способами досягнення мети вищого рівня і мають бути представлені так, щоб їх сукупність зумовлювала досягнення початкової мети [14].

Дерево цілей проекту приведено на рисунку 4.1.

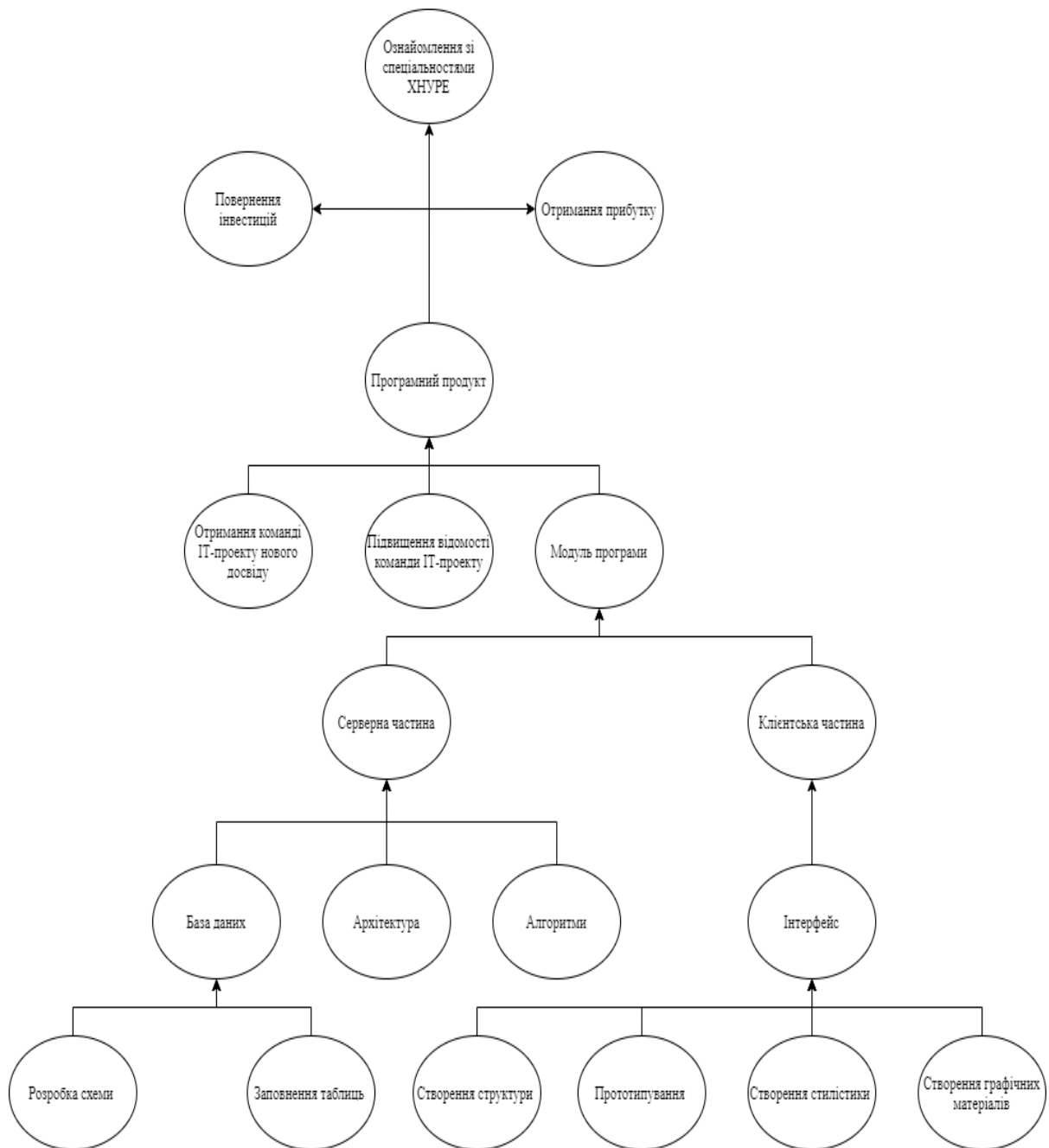


Рисунок 4.1 – Дерево цілей ІТ-проєкту

Генеральна мета проекту (мета нульового рівня) – ознайомлення зі спеціальностями ЗВО, отримання прибутку від реалізації ПП та повернення інвестицій зовнішніх інвесторів.

Під розробкою розуміється проектування ПП впродовж 237 робочих днів із вартістю проекту 100000 грн та датою початку проекту, що була призначена на 10 липня 2021 року.

Для досягнення генеральної мети варто виконати ціль першого рівня – розробити ПП. Більш детальну уяву щодо досягнення цілей першого рівня надають цілі другого рівня. Щоб досягти виконання такої цілі першого рівня, необхідно команді IT-проекту, отримати нового релевантного досвіду з розробки проектів, щоб успішно взятись за розробку цього IT-проекту, підвищити рівень відомості фахівців у команді IT-проекту, щоб інвестори довірилися їй і вклали гроші на розробку проекту, а також, необхідно безпосередньо розробити модуль програми.

Більш детальну уяву щодо досягнення цілей другого рівня надають цілі третього рівня. Щоб досягти виконання такої цілі другого рівня, як реалізації модуля програми, необхідно реалізувати серверну та клієнтську частину.

Четвертий рівень дерева цілей надає більш детальну уяву щодо робіт, які необхідно виконати за для досягання цілей третього рівня.

Для реалізації серверної частини необхідно створити базу даних, розробити архітектуру ПЗ, а саме, яким чином будуть взаємодіяти компоненти ПЗ між собою, яким чином клієнтська та серверна частина будуть обмінюватись даними, прописати алгоритми.

Для реалізації клієнтської частини необхідно створити інтерфейс ПЗ, щоб користувач зміг зручно використовувати цей мобільний додаток.

Більш детальну уяву щодо досягнення цілей надають цілі п'ятого рівня. Для створення бази даних, необхідно розробити її схему (прописати таблиці та зв'язки у третій нормальній формі й далі заповнити поля таблиці необхідними значеннями. Для створення інтерфейсу ПЗ необхідно створити

структуру інтерфейсу, його прототип, створити стилістику інтерфейсу, й далі, розробити графічний матеріал для нього.

У кожному проекті врівноважено «трикутник» обмежень – співвідношення часу, грошей, обсягів та якості робіт [14-16]. І не можна змінити один із параметрів, не зачепивши принаймні один із двох інших. Наприклад, щоб дотриматися дати завершення проекту (час), можна витратити більше ресурсів (грошей) і завершити роботу швидше або скоротити його функціональність (об'єм робіт), щоб до нового крайнього терміну об'єм робіт зменшився [14-17].

Дата початку та завершення робіт визначаються згідно графіку виконання робіт:

- дата початку проекту: 10 липня 2021 року;
- дата завершення робіт: 16 травня 2022 року.

До обмежень можна додати матеріальні, трудові та фінансові ресурси. Таким чином, бюджет проекту становить 100000 грн.

Джерелом фінансування є фонд В. Пінчука й інші зацікавлені сторони, інвестори.

Для розробки використовуються ліцензійне ПЗ: операційна система Windows; пакет програм «Office»; середовища розробки типу «AdobeEffect»; «Figma», «Android Studio»; набір сервісів для тестування; система управління базою даних (СУБД) Oracle.

4.1 Статут проекту

Статут проекту – це інструмент, який формально авторизує проект. Даний документ зазвичай випускається керівником, зовнішнім по відношенню до проекту, і наділяє менеджера повноваженнями на

використання в проекті ресурсів організації. Це особливо важливо в середовищі, де менеджери не мають безпосередньої влади над членами проектної команди та іншими ресурсами, але несуть відповідальність за виконання проекту.

Статут проекту наведено в таблиці 4.1.

Таблиця 4.1 – Статут проекту «Гід по спеціальностям закладу вищої освіти»

Назва проекту: «Гід по спеціальностям закладу вищої освіти».	
Керівник проекту:	Посада – Проектний менеджер Ярмак В.В.
Куратори проекту:	Фонд В. Пінчука.
Власник проекту:	Команда «United».
<p>Опис проекту: «Гід по спеціальностям закладу вищої освіти» – мобільний додаток, що складається з клієнтської та серверної частини.</p> <p>Серед переваг розроблюваного ПЗ можна виділити новизну підходу (централізованого) щодо пошуку здобувачів вищої освіти з певної освітньої програми ХНУРЕ для спілкування, надання релевантної інформації щодо навчання на певній освітній програмі й кафедрі, надання інформації щодо опису навичок за певною спеціальністю, що дійсно знадобились у практичній діяльності після закінчення ЗВО.</p> <p>Мета. Мобільний додаток, який дозволить професійно орієнтувати абітурієнтів, а також централізовано спілкуватись зі здобувачами вищої освіти ЗВО за різними спеціальностями, отримувати від них релевантну інформацію.</p> <p>Очікувані результати. Якісні: поліпшення процесу профорієнтації майбутніх здобувачів вищої освіти.</p> <p>Кількісні: збільшення кількості здобувачів вищої освіти, що оберуть спеціальність за інтересами та схильністю.</p>	

Продовження таблиці 4.1

<p>Допущення та обмеження:</p> <p>Строки виконання проекту дорівнюють 237 робочих дня з моменту впровадження уставу проекту.</p> <p>У разі нехватки коштів, бюджет проекту може перевищувати 100 000 грн, для цього можуть бути залучені нові інвестиції до проекту від нових зацікавлених сторін. У разі виникнення непередбачуваних ситуацій терміни виконання проекту можуть бути збільшені на 10%.</p>		
<p>Ризики:</p> <p>Можливі негативні наслідки та причини відмови від реалізації проекту:</p> <ul style="list-style-type: none"> – недостатнє початкове фінансування; – внесення великих правок щодо функціонування додатку на останній стадії життєвого циклу ІТ-проекту. 		
Фаза проекту	Основний зміст	Дата виконання
Фаза ініціації	<p>1) Дослідження предметної області;</p> <p>2) Дослідження об'єкта та аргументування на необхідність створення проекту;</p> <p>3) Формування статуту проекту;</p> <p>4) Формування вимог користувача (технічне завдання);</p>	10.06.2021 – 05.08.2021
Фаза планування	<p>5) Укладення договору про планування проекту;</p> <p>6) Розробка ескізу концепції проекту;</p> <p>7) Обговорення вузьких місць ескізу концепції;</p> <p>8) Документація про завершення планування;</p>	05.08.2021 – 10.09.2021

Кінець таблиці 4.1

Фаза розробки	9) Створення структури інтерфейсу; 10) Прототипування інтерфейсу; 11) Створення стилістики інтерфейсу; 12) Створення графічних матеріалів; 13) Підготовка матеріалів для розробника; 14) Розробка схеми БД; 15) Заповнення таблиць БД ; 16) Розробка проекту;	11.09.21 – 06.04.22
Фаза завершення	17) Тестування проекту; 18) Запуск проекту; 19) Здача проекту; 20) Укладення договору про закінчення; розробки проекту.	06.04.22 – 16.05.22
Документ підготовлено: Ярмак В.В. – здобувач вищої освіти групи УПГІТ-20-1м.		

4.2 Структуризація проекту

В ході уточнення дерева цілей була створена ієрархічна структура декомпозиції робіт (Work Breakdown Structure).

WBS IT-проекту у вигляді графа приведена на рисунку 4.2.

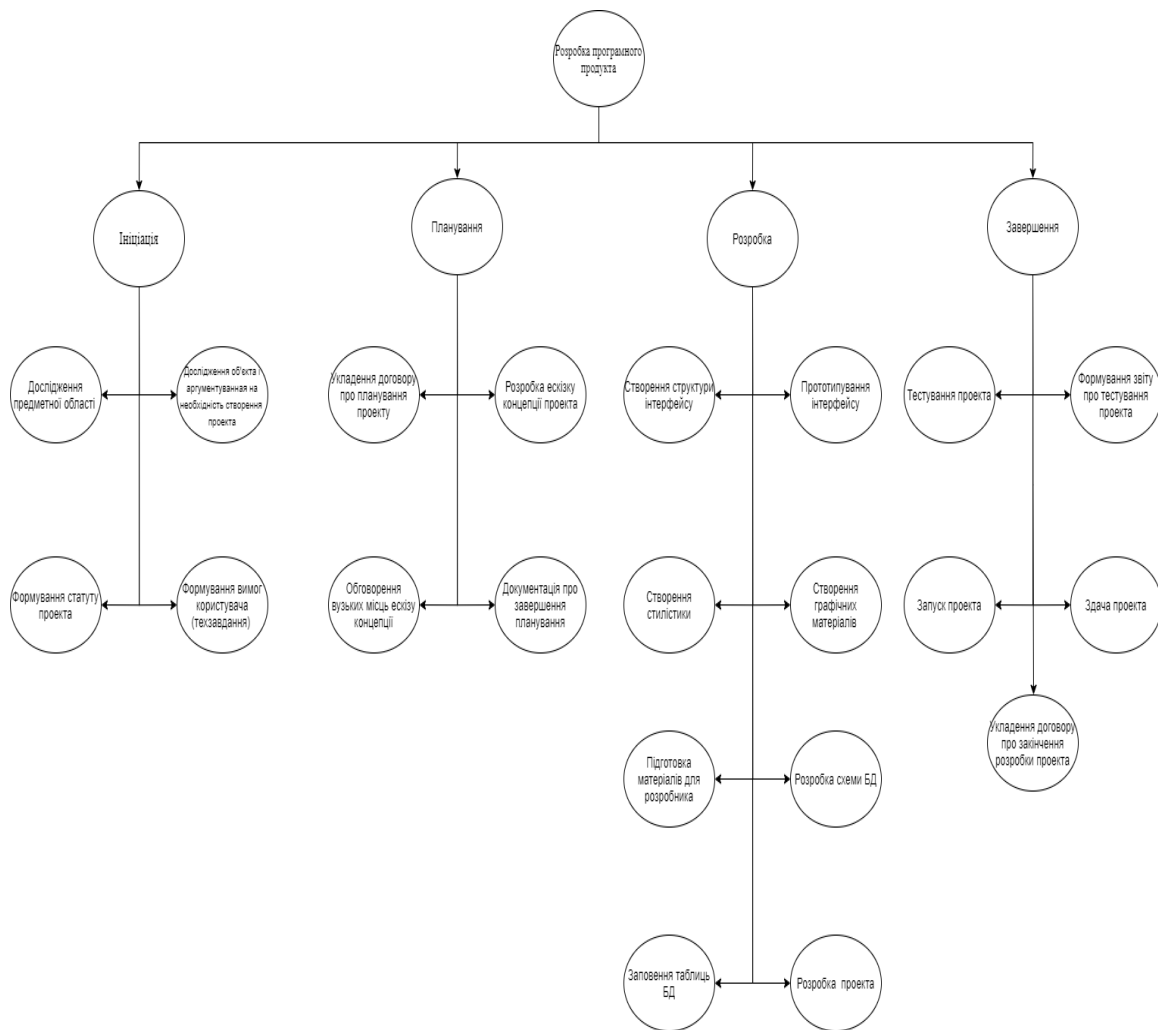


Рисунок 4.2 –WBS IT-проекту у вигляді графа

Схема організаційної структури виконавців проекту приведена на рисунку 4.3.



Рисунок 4.3 – OBS команди IT-проекту

Проектний менеджер – фахівець, що є формальним керівником команди проекту. Він відповідає за мотивацію, керування, планування та управління командою.

Бізнес-аналітик – фахівець, що відповідає за розробку та оформлення технічної документації, розмову з замовником, оптимізацію бізнес-процесів ІТ-проекту.

Дизайнер – фахівець, що розробляє стилістику, графічні матеріали, прототипи інтерфейсів ІТ-проекту.

Розробник – фахівець, що розробляє ПЗ в ІТ-проекті. Саме він реалізовує функціонал програмного забезпечення.

Тестувальник – фахівець, що тестує ПЗ. Якщо будуть знайдені помилки в ПЗ, то він доповість розробнику про них для їх подальшого виправлення.

На основі побудованих WBS та OBS створюємо матрицю відповідальності. Ролі у матриці відповідальності такі : В – виконавець; П – приймання роботи; КО – координація робіт; К – контроль виконання робіт.

Матриця відповідальності проекту приведена у таблиці 4.2.

Таблиця 4.2 – Матриця відповідальності проекту

№	Проектний менеджер	Тестувальник (Т)	Розробник (Р)	Дизайнер (Д)	Бізнес-аналітик (БА)
1	2	3	4	5	6
1.1	В, КО,П,К				В, П
1.2	КО, П, К				В

Кінець таблиці 4.2

1.3	КО, П,К				В
1.4	КО, П, К, В	П	П	П	В, П
2.1	КО, В, П, К				В, П
2.2	КО, П, К				В
2.3	КО, П, К				В
2.4	КО, П, К				В
3.1	КО, П, К			В	
3.2	КО, П, К			В	
3.3	КО, П, К			В	
3.4	КО, П, К			В	
3.5	КО, П, К		П	В	
3.6	КО, П, К		В		
3.7	КО, П, К		В		
3.8	КО, П, К	П	В		
4.1	КО, П, К	В	П		
4.2	КО, П, К	В	П		
4.3	КО, П, К		В		
4.4	КО, В, П, К				В
4.5	КО, В, П, К				В

В таблиці введені позначення : 1 – Дослідження предметної області, 2 – Дослідження об'єкта і аргументування на необхідність створення проекту, 3 – Формування статуту проекту, 4 – Формування вимог користувача (технічне завдання), 5 – Укладення договору про планування проекту, 6 – Розробка ескізу концепції проекту, 7 – Обговорення вузьких місць ескізу концепції, 8 – Документація про завершення планування, 9 – Створення структури інтерфейсу, 10 – Прототипування інтерфейсу, 11 – Створення стилістики інтерфейсу, 12 – Створення графічних матеріалів, 13 – Підготовка матеріалів для розробника, 14 – Розробка схеми БД, 15 – Заповнення таблиць БД , 16 – Розробка проекту, 17 – Тестування проекту, 18 – Формування звіту про тестування проекту , 19 – Запуск проекту, 20 – Здача проекту, 21 – Укладення договору про закінчення розробки проекту.

На основі створених вище елементи була створена структура споживаних ресурсів (Resource Breakdown Structure). Дана структура RBS зображена на рисунку 4.4.

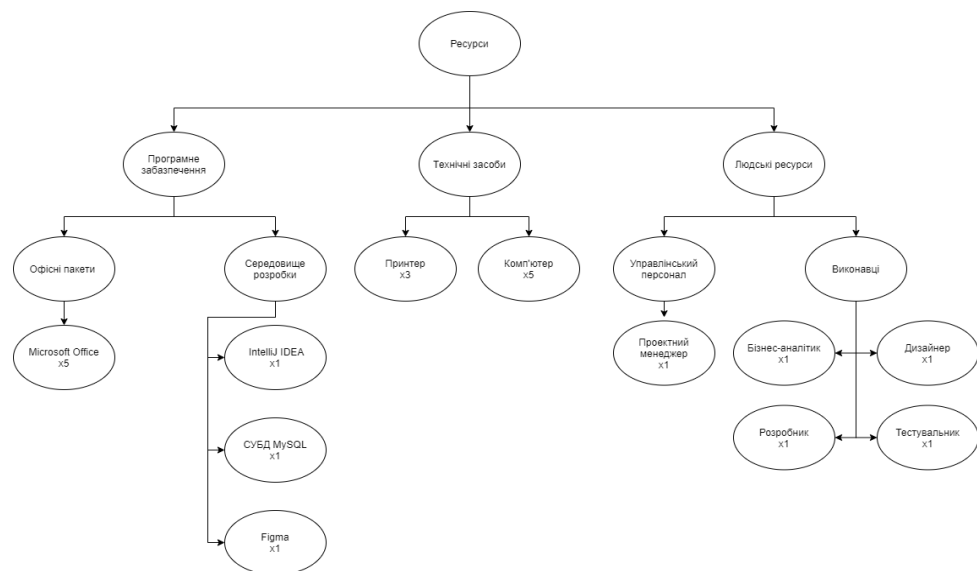


Рисунок 4.4 – Схема RBS для проекту

На основі створеної схеми RBS створюється схема Accounting Breakdown Structure (ABS).

На рисунку 4.5 зображена схема ABS.

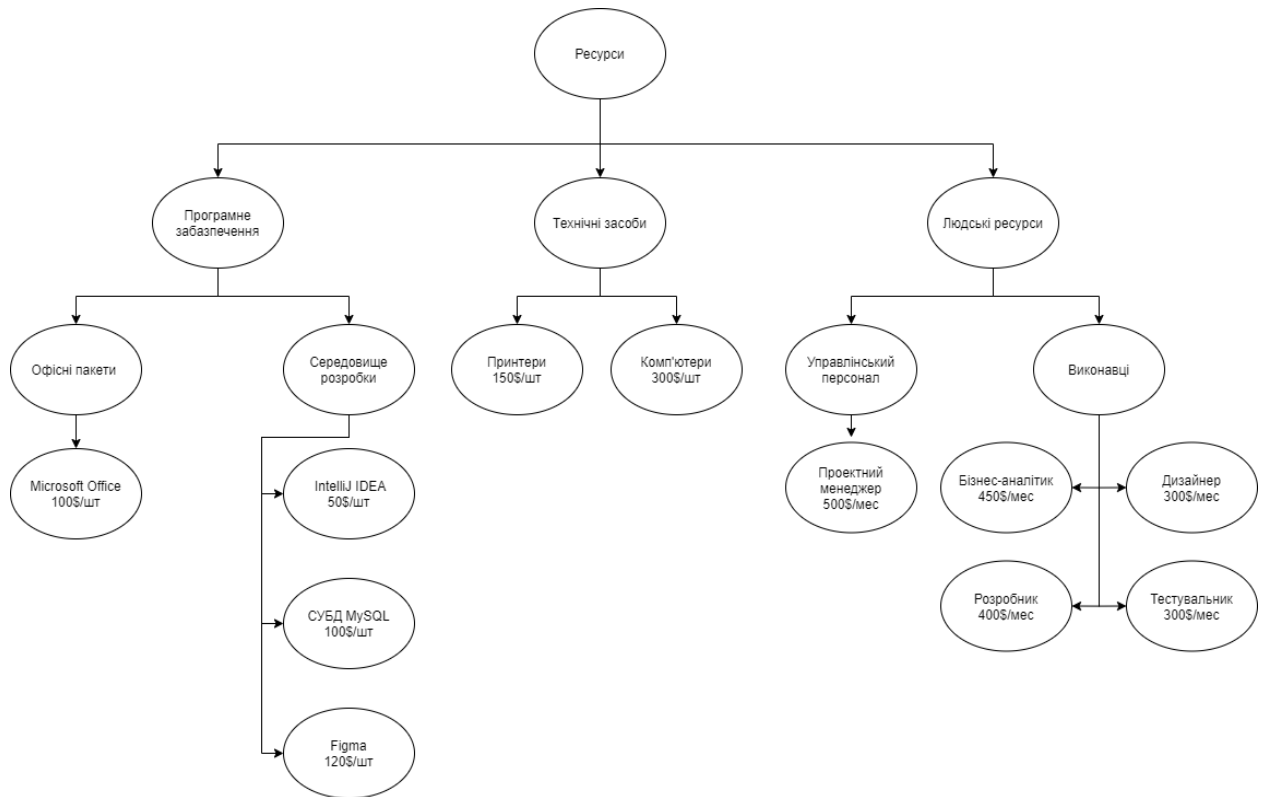


Рисунок 4.5 – Схема ABS проекту

4.3 Керування часом, вартістю та ресурсами проекту

4.3.1 Розробка моделі проекту за допомогою інформаційної технології MS Project

У MSProject було сформовано календарний план проекту, він включає перелік робіт, тривалість виконання робіт, дату початку робіт та їх кінець

WBS IT-проекту у вигляді списку наведена на рисунку 4.6.

▲ Фаза ініціації	41 днів	Чт 10.06.21	Чт 05.08.21
Дослідження предметної області	23 днів	Чт 10.06.21	Пн 12.07.21
Дослідження об'єкта і аргументування на необхідність створення проекту	5 днів	Вт 13.07.21	Чт 22.07.21
Формування статуту проекту	3 днів	Чт 22.07.21	Вт 27.07.21
Формування вимог користувача (техзавдання)	7 днів	Вт 27.07.21	Чт 05.08.21
▲ Фаза планування	26 днів	Чт 05.08.21	Пт 10.09.21
Укладення договору про планування проекту	4 днів	Чт 05.08.21	Ср 11.08.21
Розробка ескізу концепції проекту	8 днів	Ср 11.08.21	Пн 23.08.21
Обговорення вузьких місць ескізу концепції	8 днів	Пн 23.08.21	Чт 02.09.21
Документація про завершення планування	6 днів	Чт 02.09.21	Пт 10.09.21
▲ Фаза розробки	144 днів	Сб 11.09.21	Ср 06.04.22
Створення структури інтерфейсу	3 днів	Сб 11.09.21	Вт 12.10.21
Прототипування інтерфейсу	15 днів	Вт 12.10.21	Ср 03.11.21
Створення стилістики інтерфейсу	6 днів	Ср 03.11.21	Чт 11.11.21
Створення графічних матеріалів	13 днів	Чт 11.11.21	Вт 30.11.21
Підготовка матеріалів для розробника	5 днів	Ср 01.12.21	Вт 07.12.21
Розробка схеми БД	6 днів	Вт 07.12.21	Ср 15.12.21
Заповнення таблиць БД	10 днів	Ср 15.12.21	Ср 29.12.21
Розробка проекту	67 днів	Ср 29.12.21	Ср 06.04.22
▲ Фаза завершення	26 днів	Ср 06.04.22	Пн 16.05.22
Тестування проекту	8 днів	Ср 06.04.22	Пн 18.04.22
Формування звіту про тестування проекту	3 днів	Пн 18.04.22	Пт 22.04.22
Запуск проекту	7 днів	Пт 22.04.22	Вт 03.05.22
Здача проекту	2 днів	Вт 03.05.22	Ср 11.05.22
Укладення договору про закінчення розробки проекту	3 днів	Ср 11.05.22	Пн 16.05.22

Рисунок 4.6 – WBS IT-проекту у вигляді списку

Діаграма Гантта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню.

Діаграма Гантта IT-проекту зі створення мобільного додатку «Гід по спеціальностям закладу вищої освіти» представлено на рисунку 4.7.

Таблиця ресурсів проекту наведена на рисунку 4.8.

Название ресурса	Тип	Единицы измерения материалс	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных
Компьютеры	Материальный	шт	К	ИТ		\$300,00	
Принтеры	Материальный	шт	П	ИТ		\$150,00	
Пакеты Microsoft Office (Word, Excel, PowerPoint, Visio, MS Project)	Материальный	копии	П	ИТ		\$100,00	
Програми СУБД MySQL	Материальный	копии	ПСУБ	ИТ		\$100,00	
Програми IntelliJ IDEA	Материальный	копии	IdEA	ИТ		\$50,00	
Figma	Материальный	копии	Fi	ИТ		\$120,00	
Проектний-менеджер	Трудовой		П	ИТ	100%	\$500,00/мес	\$0,00/ч
Бізнес-аналітик	Трудовой		БА	ИТ	100%	\$450,00/мес	\$0,00/ч
Розробник	Трудовой		Разр	ИТ	100%	\$400,00/мес	\$0,00/ч
Дизайнер	Трудовой		Д	ИТ	100%	\$300,00/мес	\$0,00/ч
Тестувальник	Трудовой		Т	ИТ	100%	\$300,00/мес	\$0,00/ч

Рисунок 4.8 – Таблица ресурсів проекту

Як видно з таблиці найбільші заробітні плати мають проектний менеджер та бізнес-аналітик, оскільки саме вони беруть відповідальність за успішне створення ІТ-проекту, за функцію контролю та прийом отриманих результатів.

4.3.2 Оптимізація проекту за показниками час-вартість

План робіт проекту був сформовано вірно з самого початку. При плануванні перевищення крайніх термінів роботами плану проекту виявлено не було.

Формула для вираховання середньозваженої величини отримана експериментально і має такий вигляд

$$PERT_{сер} = (PERT_{оптим.} + 4 PERT_{ймов.} + PERT_{песим.})/6 \quad (4.1)$$

де $PERT_{оптим}$ – оптимістична оцінка (роботи виконуються якнайшвидше);

$PERT_{ймов}$ – найімовірніша оцінка (роботи виконуються згідно із графіком);

$PERT_{песим}$ – песимістична оцінка (робочий процес не оптимізовано, кількість непередбачуваних процесів – велика).

В даному проекті, під оптимістичною оцінкою пропонується розуміти підрахунок часу виконання критичних робіт у робочих днях, з максимально можливим розпаралелюванням. Тобто, за рахунок перерозподілу обов'язків та перевантаження ресурсів досягається виграш у часі. Тож оптимістична оцінка по методу PERT становить 190 робочих днів.

Найімовірнішою є оцінка, що становить 237 днів, оскільки у більшості випадків вона відповідає запланованій тривалості проекту.

Песимістична оцінка має на меті врахування тривалості задач, якби вони усі виконувалися послідовно, тобто максимально довго.

Використовуючи діаграму Гантта можна визначити, що при послідовному виконанні та відсутності перевантаження ресурсів, тривалість проекту становить 258 днів.

Розрахуємо середнє значення за формулою (1): $(190 + 237 \cdot 4 + 258) / 6 = 233$. Виходячи із розрахунків, середня тривалість проекту складає 233 робочих днів.

4.3.3 Планування системи управління проектом

План управління комунікаціями проекту приведений у таблиці 4.3

Таблиця 4.3 – План управління комунікаціями

Предмет комунікації	Мета комунікації	Частота	Дати початку/ завершення	Формат/ засіб зв'язку	Відповідальна особа
1	2	3	4	5	6
Договір про створення ІТ-проекту	Ознайомлення всіх зацікавлених сторін з договором про створення ІТ-проекту	На початку проекту	11.06.21 – 12.06.21	Розсилка електронної версії документу через Telegram	Проект-ний менеджер
Статут проекту	Ознайомлення всіх учасників проекту з статутом проекту	Початок кожного місяця	22.07.21 – 27.07.21	Розсилка електронної версії документу через Telegram	Проект-ний-менеджер
Технічне завдання	Ознайомлення всіх учасників з технічним завданням	Перед початком кожної роботи	11.08.21 – 23.08.21	Розсилка електронної версії документу через Telegram	Бізнес-аналітик

Кінець таблиці 4.3

1	2	3	4	5	6
Укладений договір про планування проекту	Ознайомлення бізнес-аналітика з договором про планування проекту	Перед розробкою ескізу концепції	05.08.21 – 11.08.21	Розсилка електронної версії документу через Telegram	Проект-ний менеджер
Ескіз концепції проекту	Ознайомлення проектного менеджера	Перед створенням документації про завершення планування	11.08.21 – 02.09.21	Розсилка електронної версії документу через Telegram та GitHub Repository	Бізнес-аналітик
Документація про завершення планування	Ознайомлення проектного менеджера з документацією про завершення планування	На кінці етапу планування	02.09.21 – 10.09.21	Розсилка електронної версії документу через Telegram	Бізнес-аналітик

Draw.IO – необхідний для побудови візуальних моделей вимог, та аналізу варіантів використання для знаходження дублюючих функцій.

MSProject – інструмент планування, контролю, аналізу стану проекту та робіт на протязі усього життєвого циклу проекту.

Telegram – необхідний інструмент для комунікації учасників проекту, та пересилання невеликих об’ємів даних.

GitHub Repository – необхідний інструмент комунікації, який дозволяє відстежувати, хто і коли вносив зміни до проекту, а також зберігати та обмінюватися великими об’ємами файлів.

Треба зазначити, що вхідні інформаційні потоки – це інформація з паперових носіїв, вона перетворюється на вихідні цифрові потоки у цифровому вигляді.

Концептуальна діаграма потоків даних процесу «Управління комунікаціями» та її декомпозиція представлені відповідно на рисунках 4.9 та рисунку 4.10.

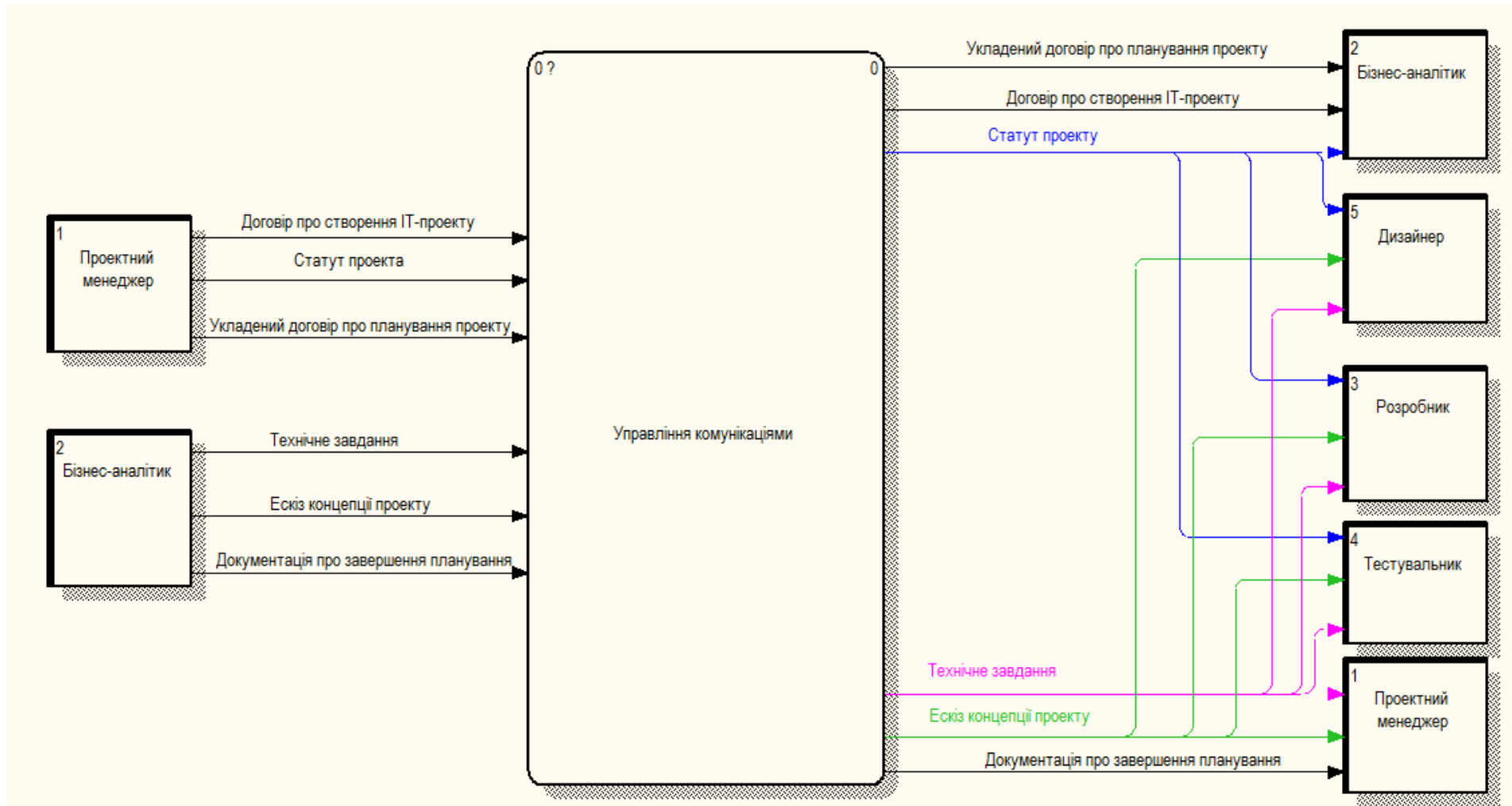


Рисунок 4.9 – Концептуальна діаграма потоків даних процесу «Управління комунікаціями»

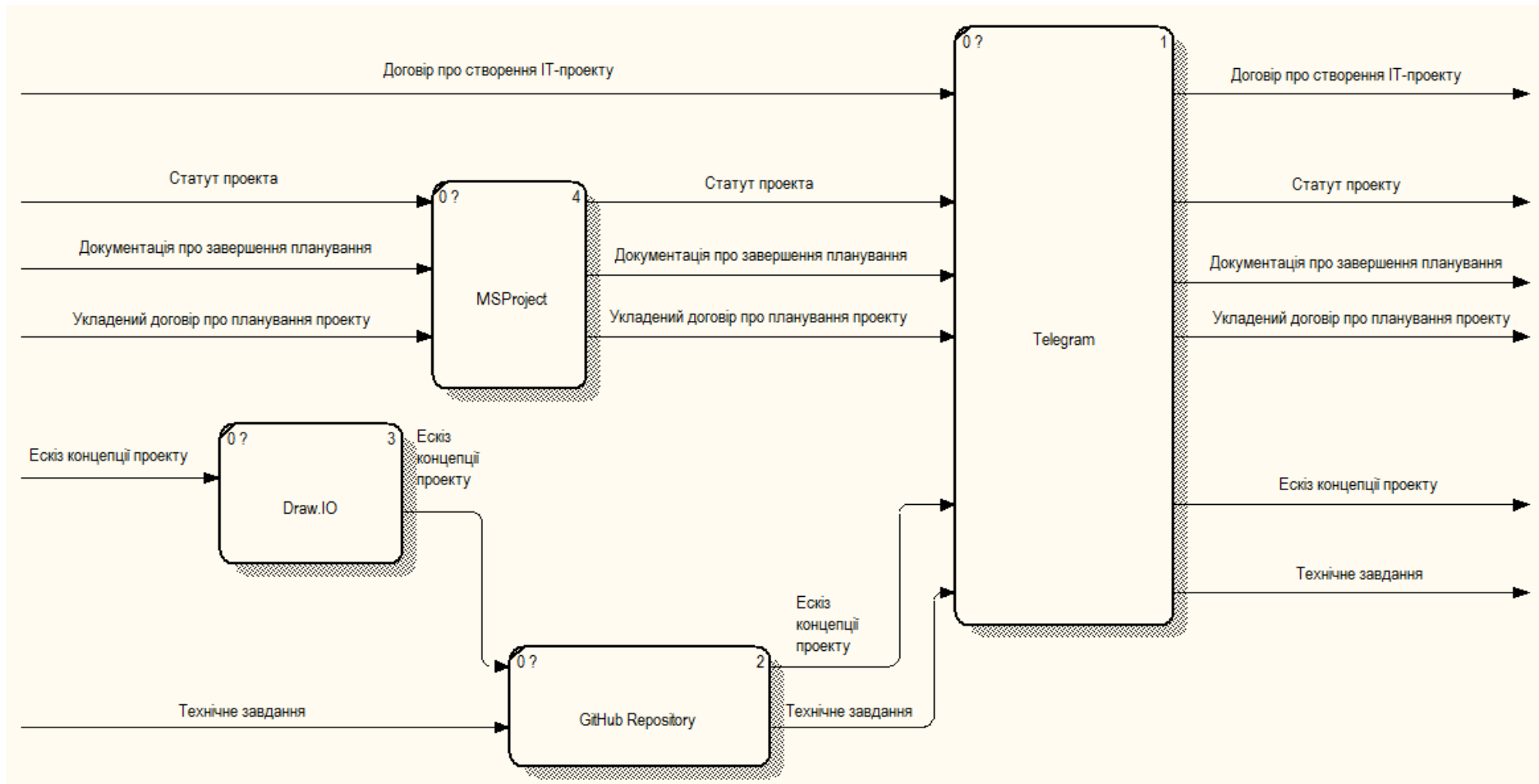


Рисунок 4.10 – Декомпозиція концептуальної діаграми потоків даних процесу «Управління комунікаціями»

4.4 Приклад використання модифікованого методу оцінки трудовитрат ІТ-проекту з розробки мобільного додатку

У прикладі буде описано метод «секторів» за участю 3 членів Scrum Team, щоб зменшити кількість ітерацій. Загальна кількість User Stories у прикладі дорівнює 20.

На рисунку 4.11 зображено початковий стан дошки з методом «секторів».

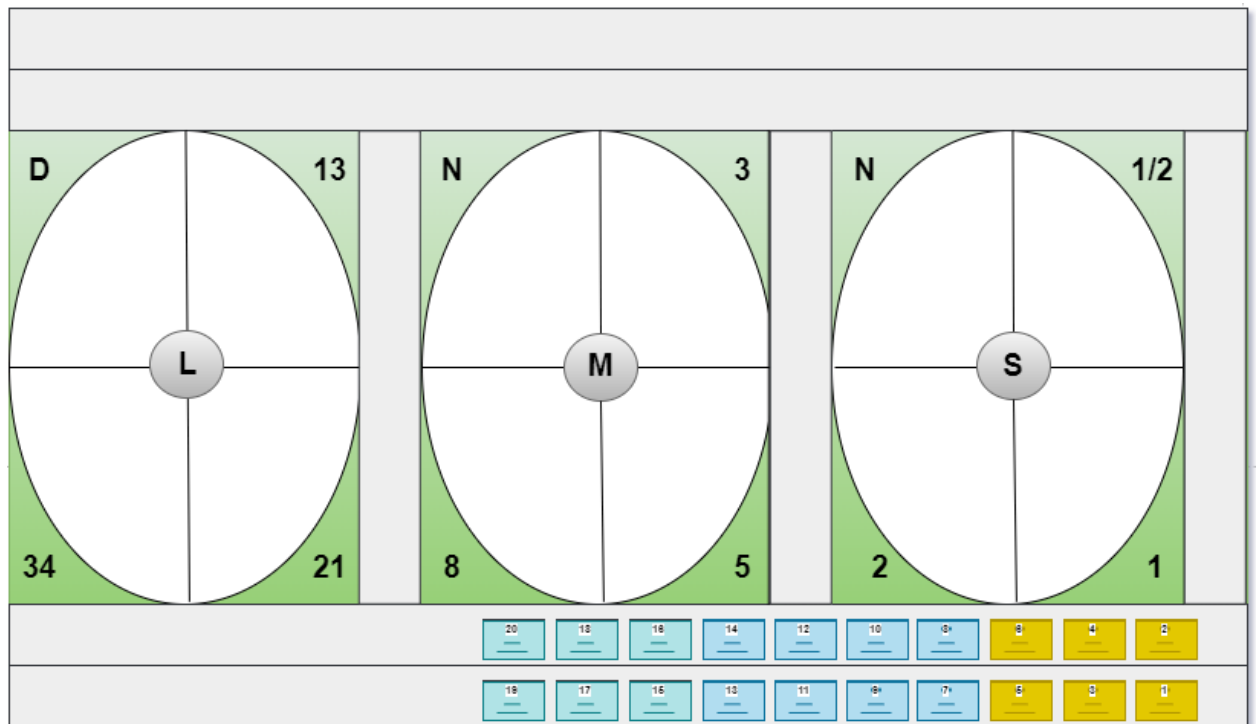


Рисунок 4.11 – Початковий стан дошки з методом «секторів»

На початковому етапі члени Scrum Team разом оцінюють перші 3-6 User Stories.

На рисунку 4.12 зображено стан дошки з методом «секторів», коли перші 4 User Stories розподілені великими мазками між трьома кругами.

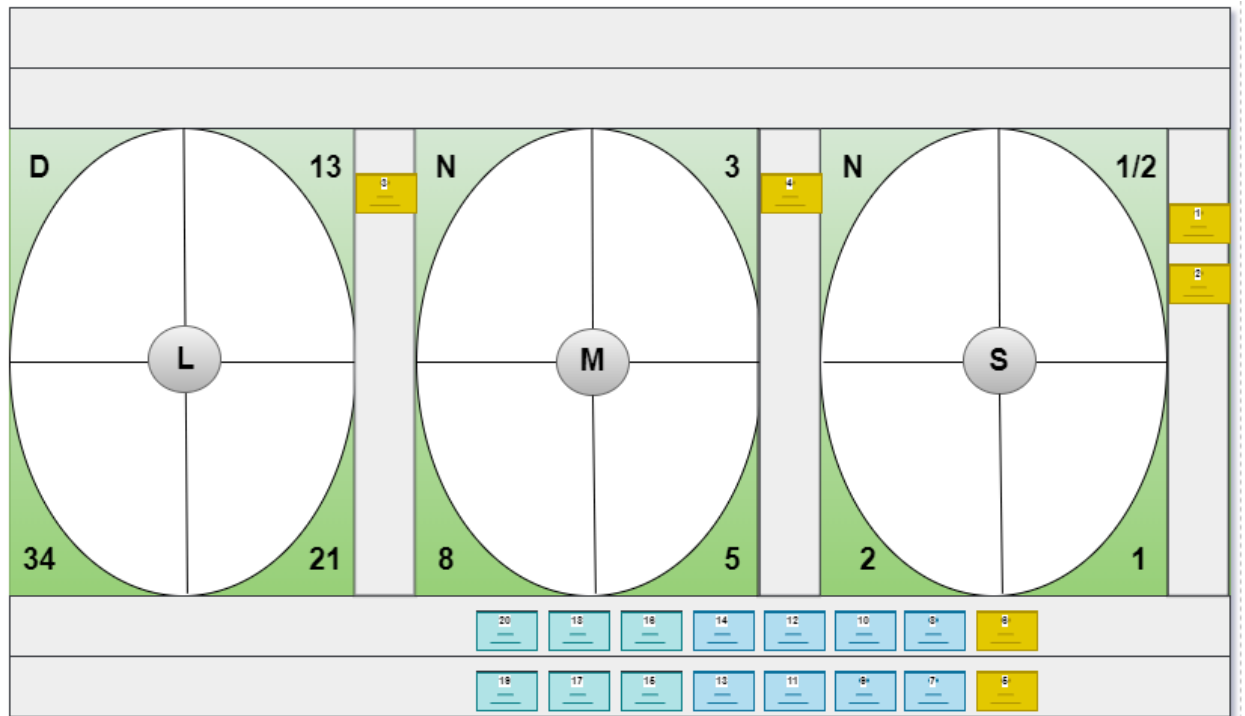


Рисунок 4.12 – Розподілення перших 4 User Stories між трьома колами

Далі, загальна кількість User Stories, що залишилась після орієнтовного розподілення деякої кількості перших User Stories розподіляється рівномірно між усіма членами Scrum Team й вони паралельно розподіляють User Stories великими «мазками» серед беклогів кіл S, M та L.

Якщо ж, кількість, що залишилась не ділиться рівномірно між усіма членами Scrum Team, тоді User Stories будуть розподілені за остаточним принципом.

В нашому випадку, було 20 User Stories, після першого орієнтовного розподілення Scrum Team залишилось 16. $16/3 = 5.3$. Перший член команди буде мати 6 User Stories. другий – 5, третій – 5.

Незрозумілі User Stories Scrum Team розташовує у верхньому беклогу для подальшої декомпозиції (верхня частина дошки).

На рисунку 4.13 зображені розподіленні User Stories великими мазками усіма членами Scrum Team між беклогами трьох кіл – S, M та L, а також ті User Stories, що необхідно додатково уточнити або декомпонувати в подальшому.

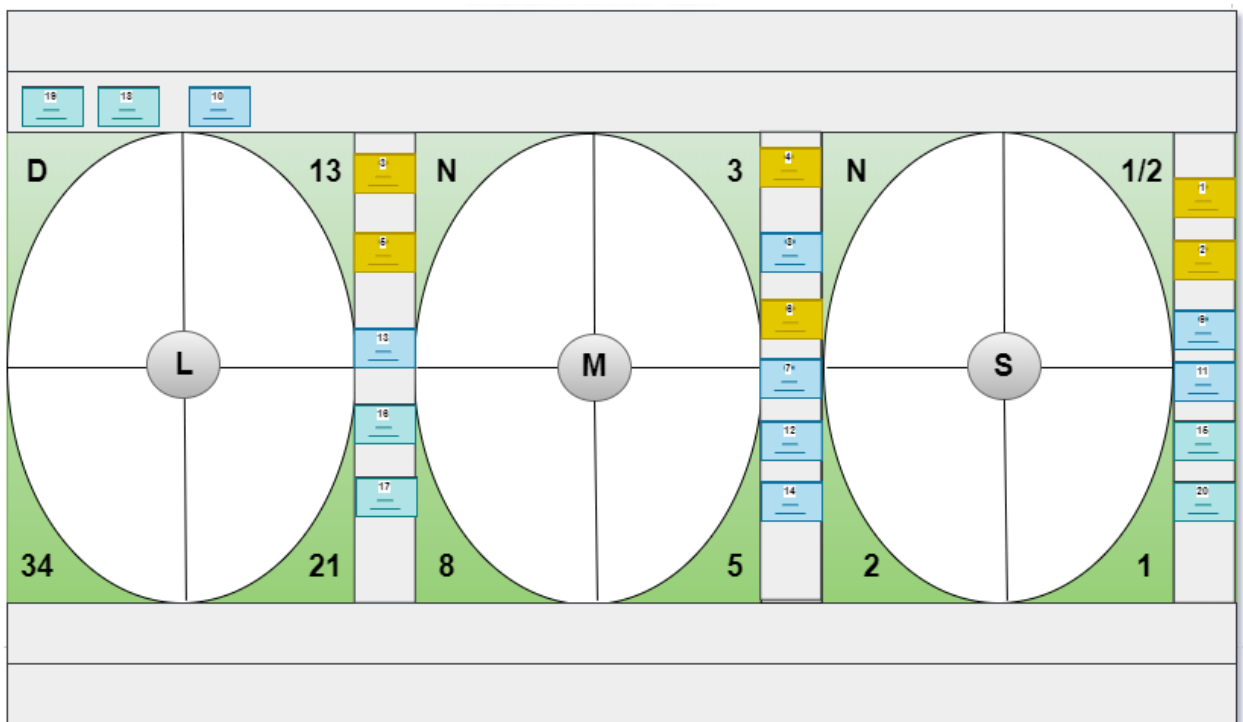


Рисунок 4.13 – Розподілення User Stories великими мазками

Далі, необхідно поетапно розподілити User Stories в кожному з секторів. Кожен член команди буде покроково оцінювати User Stories у колах S, M та L. Спочатку, члени Scrum Team починають оцінювати User Stories з S, далі переходять у коло M, а далі у коло L.

Таким чином, дана послідовність дій повторюється до тих пір, коли кожен з членів команди починає пропускати свій хід. Це означає, що всі члени команди згодні з тим, яким чином розташовані User Stories у секторах певного кола.

На рисунку 4.16 зображено розподіленні User Stories третім членом команди секторах кола S.

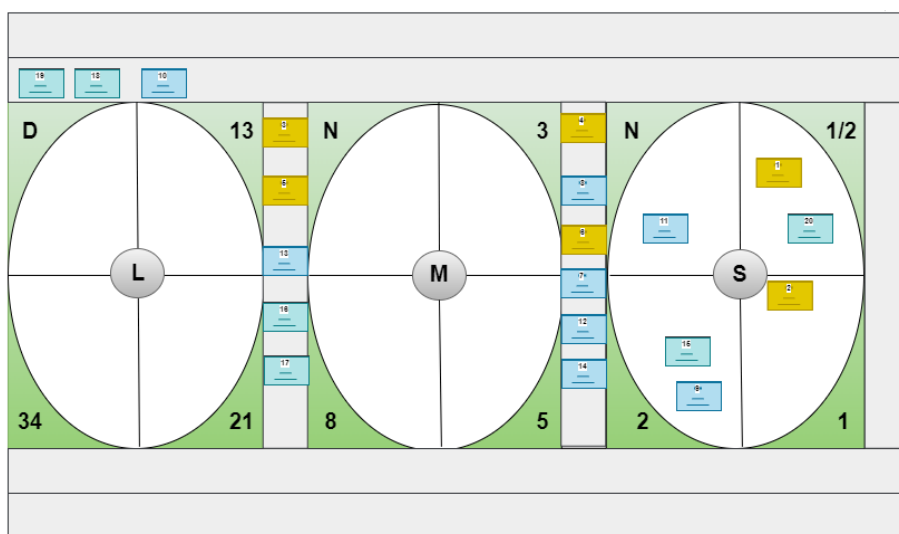


Рисунок 4.16 – Розподілення User Stories третім членом Scrum Team у секторах кола S (1 ітерація)

На рисунку 4.17 зображено розподіленні User Stories після всіх ітерацій членів Scrum Team між секторами кола S.

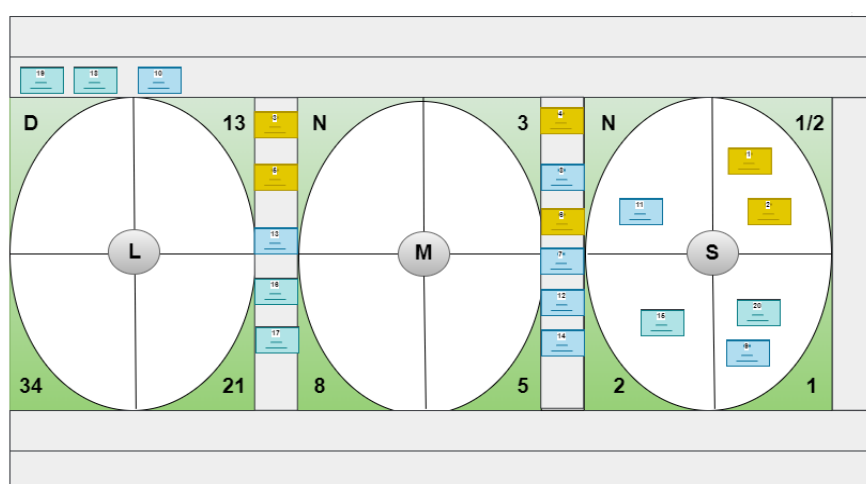


Рисунок 4.17 – Розподілення User Stories після всіх ітерацій членів Scrum Team між секторами кола S

Зазначимо, що User Story № 11 перекладається у беклог кола М.

На рисунку 4.18 зображено як User Stories , що потрапили у сектор N кола S переміщуються у кінець беклога кола М.

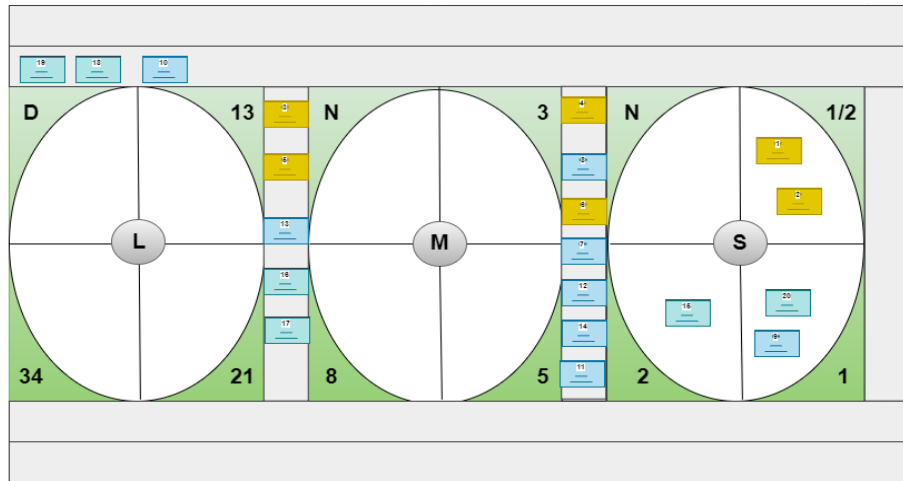


Рисунок 4.18 – Переміщення User Stories з сектора N кола S у кінець беклога кола М

На рисунку 4.19 можна побачити, повторення послідовності дій розподілення User Stories у секторах кола М після всіх ітерацій.

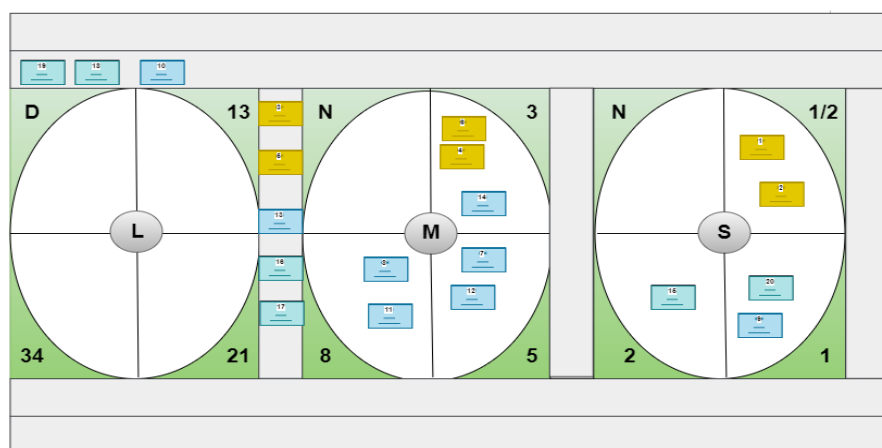


Рисунок 4.19 – Повторення послідовності дій розподілення User Stories між членами Scrum Team у секторах кола М після всіх ітерацій

На рисунку 4.20 можна побачити, повторення послідовності дій розподілення User Stories у секторах кола L після всіх ітерацій.

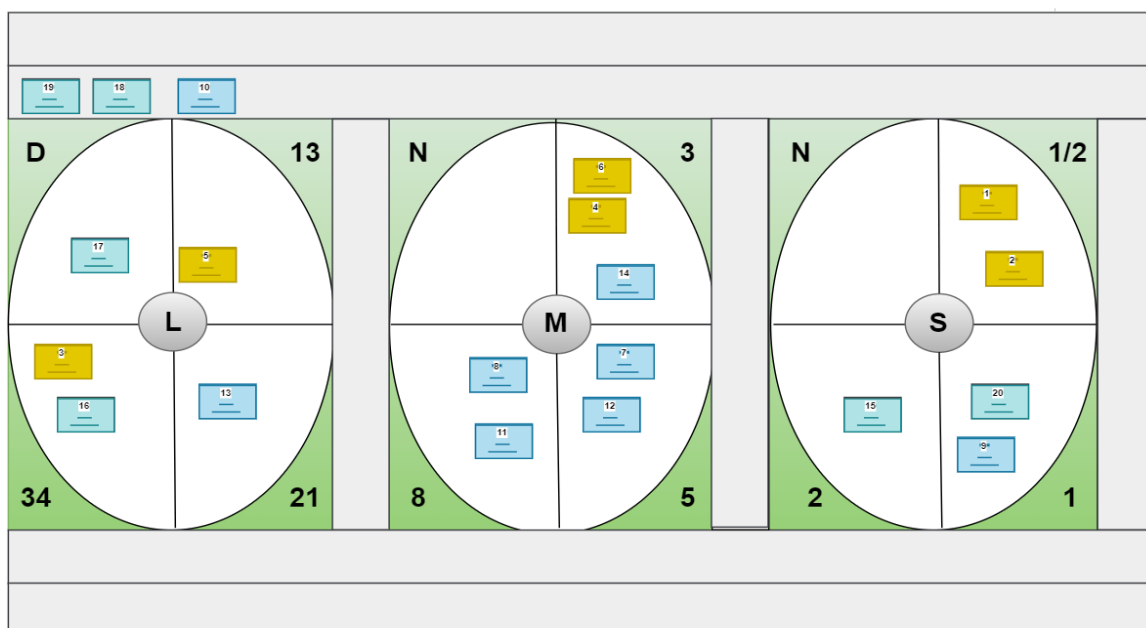


Рисунок 4.20 – Повторення послідовності дій розподілення User Stories членами Scrum Team у секторах кола L після всіх ітерацій

Як можна побачити, User Story № 17 виявилось занадто складним, щоб дати оцінку складності до 34 User Points. Цю User Story необхідно буде декомпонувати.

На рисунку 4.21 можна побачити декомпозицію всіх User Stories, яких треба декомпонувати.

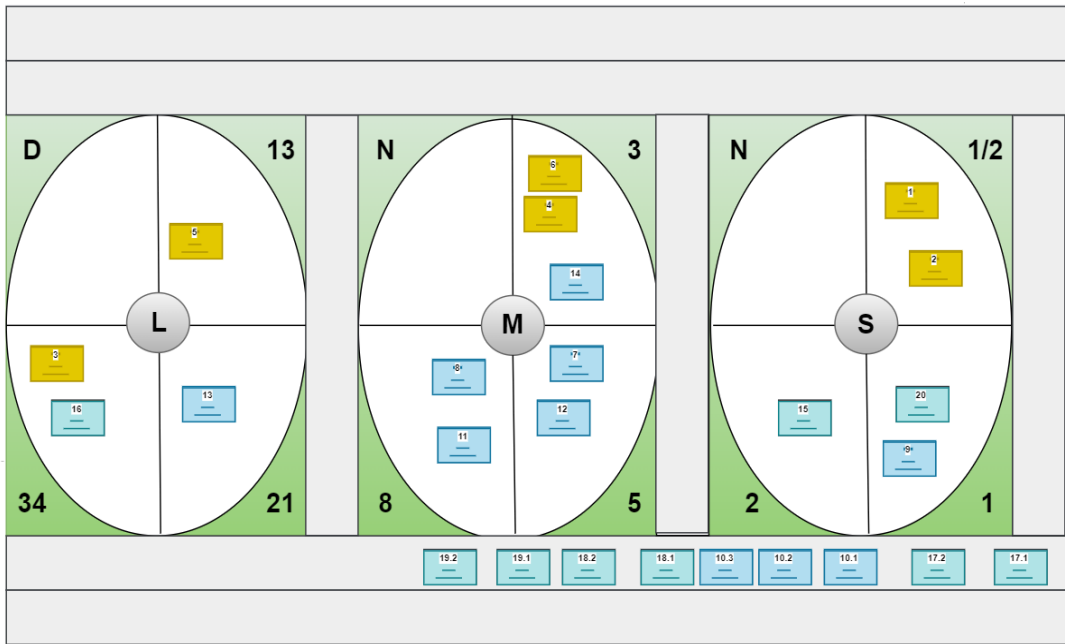


Рисунок 4.21 – Декомпозиція всіх User Stories, які треба декомпонувати

На рисунку 4.22 можна побачити розподілення декомпованих User Stories між беклогами кругів S,M,L.

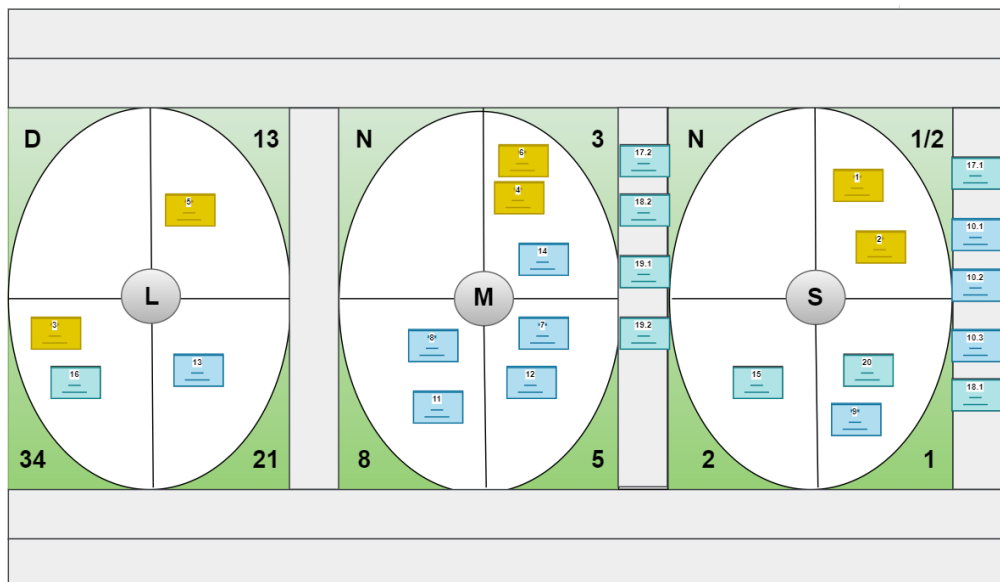


Рисунок 4.22 – Розподілення User Stories між беклогів кругів L, M,S

На рисунку 4.23 можна побачити розподілення декомпованих User Stories у секторах кола S

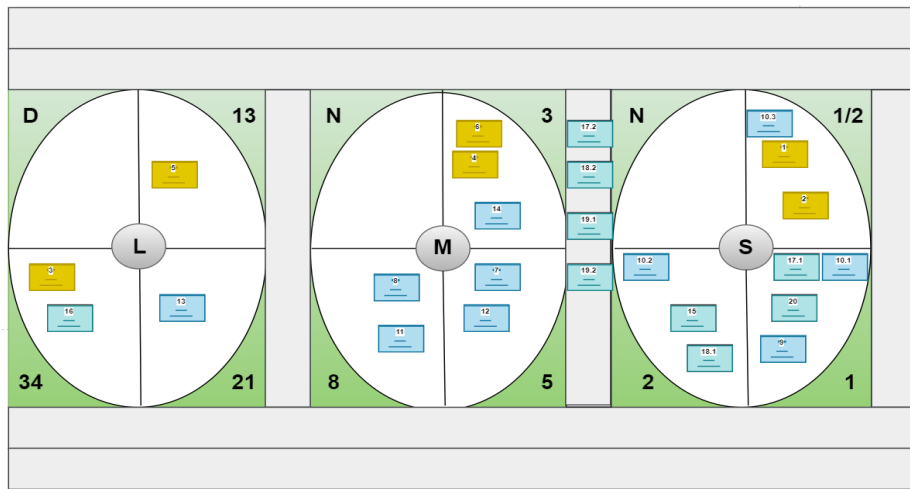


Рисунок 4.23 – Розподілення декомпованих User Stories у секторах кола S

На рисунку 4.24 можна побачити розподілення декомпованих User Stories у секторах кола M

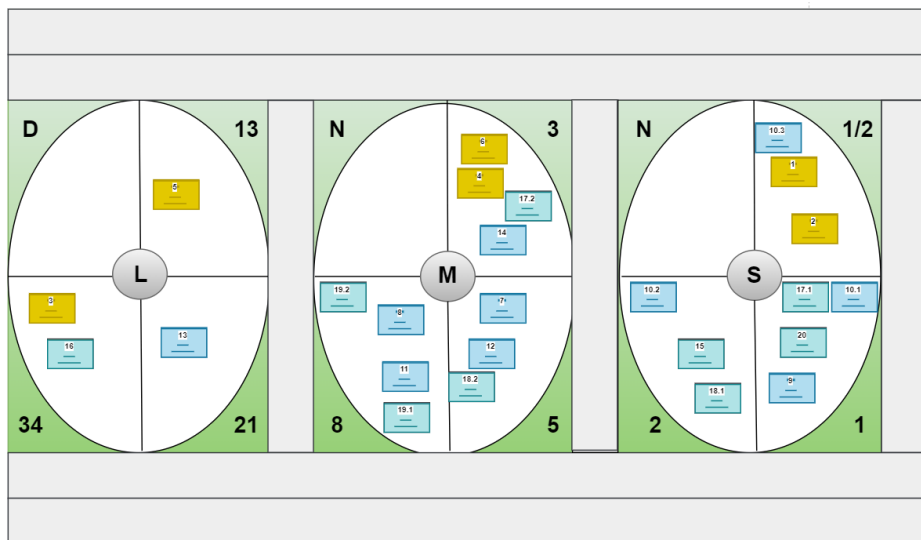


Рисунок 4.24 – Розподілення декомпованих User Stories у секторах кола M

Оскільки до беклога кола L не була потраплена жодна декомповована User Story, то розподілення у сектори кола L не було.

На рисунку 4.25 можна побачити відсортовані та оцінені User Stories, розташовані у вихідному беклогу.

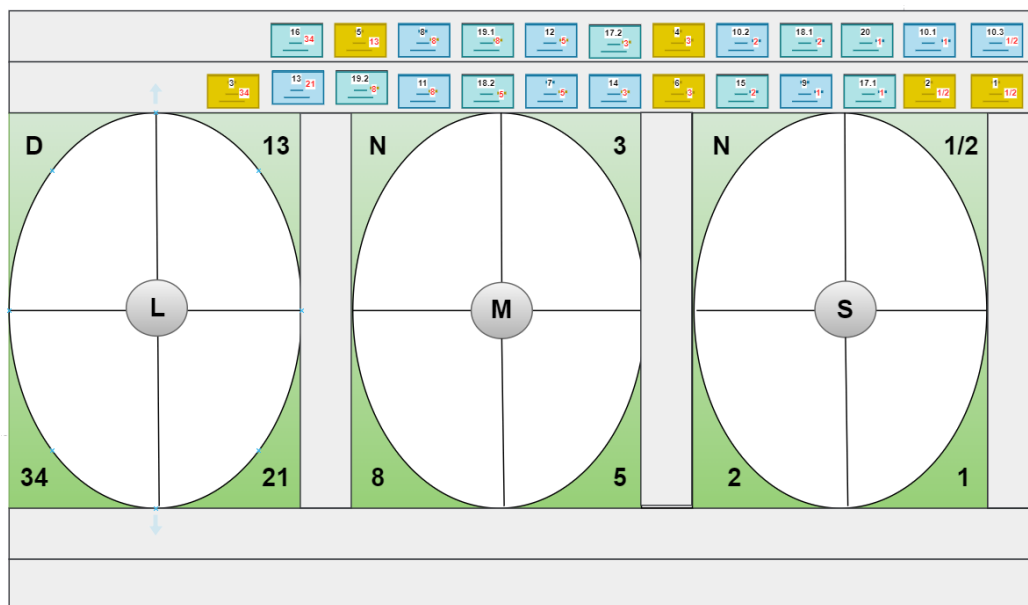


Рисунок 4.25 – Відображення відсортованих та оцінених User Stories у вихідному беклогу

Загальна кількість Stories Points для всіх Stories Point = 172,5.

Таким чином, після використання методу «секторів», Scrum Team доволі швидко та точно оцінила доволі велику кількість User Stories за доволі невеликий час.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було створено метод оцінки трудовитрат ІТ-проекту з розробки мобільного додатка, який є більш ефективний ніж існуючі методи, доволі швидкий, та легкий у використанні.

У першому розділі «Аналіз існуючих методів оцінки трудовитрат», були здійснені роботи з встановлення актуальності дослідження, проведено аналізи популярних існуючих методів та моделей оцінювання трудовитрат ІТ-проектів, виділена і сформована мета розробки модифікованого методу оцінювання трудовитрат ІТ-проектів з розробки мобільних додатків.

У другому розділі «Розробка модифікованого методу «секторів» на підставі отриманих даних досліджень були коротко описані загальні принципи модифікованого методу «секторів», коротко описані етапи модифікованого методу, приведена зведена таблиця з використанням етапів та описом модифікацій похідних методів у модифікованому методі «секторів», ретельно описана модифікація кожного з етапів методу «секторів».

У третьому розділі «Опис практичного вирішення задачі оцінювання трудовитрат розробки ІТ- проекту з розробки мобільного додатку» приведено алгоритм реалізації модифікованого методу та інструкція членам Scrum Team з виконанням розробленого методу.

У четвертому розділі «Апробація методу секторів під час оцінювання трудовитрат ІТ-проекту з розробки мобільного додатку» приведено опис проекту, на прикладі якого використався модифікований метод, а також опис використання методу на конкретному прикладі, в якому наочно видно працеспроможність методу в умовах, близьких до реальних.

Основною сферою застосування даного методу є ІТ-компанії. Даний метод надає змогу оцінити доволі велику кількість робіт за доволі короткий проміжок часу, до того ж, він доволі швидкий й простий у використанні.

Також, можна зазначити, що даний метод є доволі універсальний, саме тому, він може використовуватись у різних сферах, для оцінки трудовитрат виконання якоїсь справи.

До недоліків модифікованого методу можна віднести той момент, що у порівнянні з існуючими інтуїтивними методами оцінки трудовитрат задач даний метод потребує більше часу для оцінювання.

Далі, є сенс створити програмний додаток для даного методу, щоб можна було зручно його використовувати онлайн людям одразу з різних куточків світу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Інформаційні управляючі системи та технології») / 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, 2021. – 25 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.
3. Розробка мобільних додатків від А до Я: повний гайд. [Електроний ресурс] // Dan-it. – 2020. – Режим доступу до ресурсу: <https://dan-it.com.ua/blog/razrabotka-mobilnyh-prilozhenij-ot-a-do-ja-polnyj-gajd/> .
4. Етапи розробки мобільних додатків [Електроний ресурс] // azoft.ru – 2020. – Режим доступу до ресурсу <https://www.azoft.ru/blog/app-development>.
5. Типи мобільних додатків [Електроний ресурс] // Qastart – 2021. – Режим доступу до ресурсу: <https://qastart.by/mainterms/59-tipy-mobilnykh-prilozhenij> (дата звернення 23.10.2021).
6. Менеджмент в ІТ [Електроний ресурс] // Doitsmartly – 2021. – Режим доступу до ресурсу: <https://doitsmartly.ru/all-articles/management/99-agile/115-agile-estimation-techniques.html> (дата звернення 25.10.2021).
7. Моделі, методи та засоби оцінки вартості програмного забезпечення [Електроний ресурс] // Dspace – 2015. – Режим доступу до ресурсу: <http://dspace.nbuu.gov.ua/bitstream/handle/123456789/1541/36-Sidorov.pdf>.
8. Евланов, М.В. Унификация методов оценивания затрат на создание современных информационных систем [Текст] / М.В. Евланов, Е.И. Соловьева // Вісник Кременчуцького національного університету ім. Михайла Остроградського. – 2014. – Випуск 5/2014 (88). – С. 62-67.

9. COCOMO II Model Definition Manual [Електроний ресурс] // «Center for Systems and Software Engineering». – Режим доступу до ресурсу: ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf.

10. COCOMO II – Constructive Cost Model [Електроний ресурс] // Сайт CSSE USC. – Режим доступу до ресурсу: <http://csse.usc.edu/tools/COCOMOII.php>. (дата звернення 26.10.2021).

11. Як ми шукали наш ідеальний інструмент для планування [Електроний ресурс] // Vc – 2015. – Режим доступу до ресурсу: <https://vc.ru/services/216490-kak-my-iskali-nash-idealnyy-instrument-dlya-poker-planirovaniya> (дата звернення 26.10.2021).

12. Ярмак В.В «Дослідження моделей і методів оцінки трудовитрат ІТ-проекту з розробки мобільних додатків». В зб.: «РАДІОЛЕКТРОНІКА ТА МОЛОДЬ В ХХІ СТОЛІТТІ ». тез. докл ХХV Міжнародного молодіжного форуму м. Харків, 2021р. С 62-63.

13. ГОСТ 19.701-90 ЕСПД. Єдина система програмної документації. Схеми алгоритмів, програм даних і систем. Умовні позначення і правила виконання. – Введ. 1992-01-01.- М.: Державний стандарт союзу РСР, 1992. – 20 с.

14. . Кліффорд Г. Управління проектами : / Грей Кліффорд, Ерік Ларсон. – К.: 2003 – 872 с.

15. Катренко А.В. Управління ІТ-проектами. – Львів: «Новий Світ – 2000», 2019. – 550 с.

16. Петренко Н.О. Управління проектами / Н.О. Петренко., Л.О. Кустрич, М.О. Гоменюк: навч. посіб.; рекомендовано МОН / Уманський нац. ун-т садівництва. – К.: ЦУЛ, 2017 – 242 с.

17. Проектні обмеження, трикутна модель обмежень. URL: <https://support.office.com/en-us/article/The-project-triangle-8c892e06-d761-4d40-8e1f-17b33fdcf810?ui=en-US&rs=en-US&ad=US> (дата звернення: 07.01.2021).