НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

"На правах рукопису"
УДК 519.87

До захисту допущено
В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

“ ____ ” _____ 20 *20* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за освітньо-професійною програмою

«*Інформаційні управляючі системи та технології*»

зі спеціальності 126 «*Інформаційні системи та технології*»

на тему:

«Інформаційна система планування ресурсів ІТ-проєктів»

Виконала:
студентка VI курсу, групи **ІС-92мп**
Мартинюк Юлія Юріївна _____

Керівник:
доцент, к.т.н., доцент,
Сперкач Майя Олегівна _____

Консультант:
професор, д.т.н., доцент,
Жаріков Едуард В’ячеславович _____

Рецензент:
доцент, к.т.н., доцент,
Писаренко Андрій Володимирович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління

Рівень вищої освіти – *другий (магістерський)*

Спеціальність – 126 «*Інформаційні системи та технології*»

Освітньо-професійна програма *«Інформаційні управляючі системи та технології»*

В.о.завідувача кафедри

_____ Олександр ПАВЛОВ

«__» _____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Мартинюк Юлії Юріївні

1. Тема дисертації «Інформаційна система планування ресурсів ІТ-проєктів», науковий керівник дисертації Сперкач Майя Олегівна, к.т.н., доцент, затверджені наказом по університету від «26» жовтня 2020 р. № 3132-с

2. Строк подання студентом дисертації “ *02* ” *12* 20 *20* р.

3. Об’єкт дослідження є процес планування ресурсів ІТ-проєктів.

4. Перелік завдань, які потрібно розробити:
   - *виконати аналітичний огляд та провести детальний аналіз існуючих систем;*
   - *розробити алгоритми розв’язання задачі планування ресурсів;*
   - *дослідити ефективність розроблених алгоритмів;*
   - *розробити програмну реалізацію інформаційної системи планування ресурсів;*
   - *виконати експериментальне дослідження розробленої системи.*

5. Орієнтовний перелік графічного (ілюстративного) матеріалу

Плакат 1 Функціонально-логічна структура програмного забезпечення
Плакат 2 Декомпозиція цілей системи
Плакат 3 UML-діаграма варіантів використання
Плакат 4 Схема обробки інформації у програмному забезпеченні

Плакат 5 UML-діаграма класів
Плакат 6 Результати експериментів (1)
Плакат 7 Результати експериментів (2)

6. Орієнтовний перелік публікацій

Інформаційна система планування ресурсів IT-проєктів / Мартинюк Ю.Ю., Сперкач М.О. // П'ятнадцята міжнародна науково-практична конференція МОДС 2020, 29 червня – 01 липня 2020 р., м. Чернігів – с.363-366.
Мартинюк Ю.Ю. Інформаційна система планування ресурсів IT-проєктів / Ю.Ю. Мартинюк, М.О. Сперкач // Матеріали V всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020 р. – с.154-159.

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання "*01*" *вересня* 20*20* р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	*Аналіз результатів огляду існуючих рішень*	*24.08*	
2	*Порівняльний аналіз існуючих методів розв'язання задачі планування ресурсів*	*1.09*	
3	*Постановка змістовна та математичної моделі задачі планування ресурсів*	*10.09*	
4	*Модифікація існуючих методів розв'язання задачі планування ресурсів*	*25.09*	
5	*Розробка інформаційної системи планування ресурсів IT-проєктів*	*15.10*	
7	*Проведення експериментальних досліджень розроблених алгоритмів*	*27.10*	
8	*Оформлення документації*	*10.11*	
9	*Подання роботи на попередній захист*	*20.11*	
10	*Подання роботи на основний захист*	*02.12*	

Студент _____ *Юлія МАРТИНЮК*

Науковий керівник _____ *Майя СПЕРКАЧ*

РЕФЕРАТ

Магістерська дисертація: 109 с., 24 рис., 23 табл., 81 джерело, 1 додаток.

Актуальність. На сьогодні стрімко зростає рівень складності об'єктів автоматизації підприємств різних сфер діяльності. Це пояснюється тим, що замовники висувають все більш специфічні вимоги до інформаційних систем для задоволення потреб підприємства. На цей час ІТ-галузь є однією з перспективніших сфер людської діяльності. Невід’ємною складовою якої є реалізація різних ІТ-проєктів від замовників з усього світу. Успішне виконання проєктів залежить від їх ефективного управління.

Успішність проєкту залежить від низки показників, що визначають його сутність та впливають на стан проєкту протягом його життєвого циклу. Ця сукупність елементів проєкту по суті є об'єктами управління.

Історія різноманітних підходів до питання управління проектами має вже п'ять тисяч років за своєю тривалістю, хоча перша згадка про управління проектами була наведена ще на початку 20 століття.

Основоположниками управління проєктами можна назвати таких трьох вчених:

Генрі Гант (1861–1919) — американський інженер, який у 1910 році створив та впровадив підхід календарного планування, де основою побудови діаграм стали горизонтальні елементи. Власне, його винахід такого планування пізніше стали називати діаграмою Ганта.

Фредерік Уінслоу Тейлор (1856–1915) – американський інженер, математик та фізик, його роботи і досі використовують як прототипи сучасних різних інструментів, куди також входить ієрархічна структура його робіт.

Анрі Файоль (1841–1925) — французький інженер, засновник головної класичної моделі управління проєктом. Став доволі відомим завдяки тому, що саме він описав 5 головних функцій управління (менеджменту), які, власне, і стали основою теорії управління проєктами.

Теоретичні аспекти досліджень щодо управління проєктами містяться у працях таких українських та зарубіжних вчених, як, зокрема О.О. Куліних, С.Д. Бушуєв, Ю.С. Грисюк, І.В. Кононенко, В.І. Максимова, В.Б. Силова, Є.К. Корноушенко, О.Г. Тімінський, В.І. Прангішвілі, Долорес Шервуд Стайгер, Ю.М. Теслі, Паула Мартін, Карел Тейт, Л.А. Заде, Деніз Колонна д'Істріа. У своїх працях вони висвітлювали питання розробки інформаційних технологій в проєктному менеджменті.

Причиною актуальності ефективного процесу планування є необхідність збільшення людського потенціалу для вирішення унікальних задач. Доволі часто компанії реалізовують одночасно одразу декілька проектів, в яких задіяні одні й ті самі людські чи технічні ресурси. Автоматизація цього процесу може збільшити ефективність роботи компанії в цілому. На сьогоднішній день існує безліч прикладів підвищення ефективності, шляхом автоматизації монотонної ручної роботи.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів» (№ ДР 0117U000919).

Метою дослідження є підвищення ефективності виконання ІТ-проєктів в компанії, за рахунок створення інформаційної системи планування ресурсів ІТ-проектів, в основу якої покладені методи, що дозволяють мінімізувати час реалізації проєкту.

Для досягнення поставленої мети необхідно виконати наступні **завдання:**

– виконати аналітичний огляд та провести детальний аналіз існуючих систем планування ресурсів, управління проєктами, керування обмеженнями та ризиками у рамках проєкту, методів планування розкладів виконання завдань у межах ресурсних обмежень;

– розробити алгоритми розв’язання задачі підтримки роботи системи планування ресурсів ІТ-проєктів;

– дослідити ефективність розроблених алгоритмів;

– розробити програмну реалізацію інформаційної системи планування ресурсів ІТ-проєктів;

– виконати експериментальне дослідження розробленої інформаційної системи, методів розв’язання задачі та ефективність використаних алгоритмів.

Об’єктом дослідження є процес управління ІТ-проєктами.

Предметом дослідження є методи планування ресурсів ІТ-проектів.

Методи дослідження. Для виконання поставлених завдань у роботі було використано методи: системного аналізу (при проектуванні інформаційної системи); теорії розкладів, дослідження операцій, теорії складності (при розробленні методів розв’язання задач розподілу ресурсів); комп’ютерного моделювання (при експериментальному дослідженні ефективності методів розв’язання задач розподілу ресурсів).

Наукова новизна одержаних результатів полягає у побудові методу планування ресурсів, з урахуванням їх обмеженності, ризиків та аналізу; розробці та вдосконаленню алгоритмів розв’язання задачі за допомогою виконання експериментальних досліджень.

Публікації. Матеріали роботи опубліковані у збірнику П’ятнадцятої міжнародної науково-практичної конференції МАТЕМАТИЧНЕ ТА ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ СИСТЕМ (МОДС 2020), а також в матеріалах V всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020).

УПРАВЛІННЯ ПРОЄКТАМИ, SCRUM, БЕКЛОГ, СПРИНТ, РЕСУРС, ТЕОРІЯ РОЗКЛАДІВ, МІНІМІЗАЦІЯ ЧАСУ ВИКОНАННЯ ЗАВДАНЬ

ABSTRACT

Master's dissertation: 109 pp., 24 figs., 23 tables, 81 sources, 1 appendix.

Topicality. Today, the level of complexity of automation facilities of enterprises in various fields of activity is growing rapidly. This is due to the fact that customers are increasingly specific requirements for information systems to meet the needs of the enterprise. Currently, the IT industry is one of the most promising areas of human activity. An integral part of which is the implementation of various IT projects from customers around the world. Successful project implementation depends on their effective management.

The success of a project directly depends on a number of indicators that determine its essence and affect the state of the project during its life cycle. This set of project elements are essentially objects of management.

The history of various approaches to project management is five thousand years old, although the first mention of project management was given in the early 20th century.

The founders of project management can be called the main three scientists:

Henry Gant (1861–1919) was an American engineer who in 1910 created and implemented a calendar planning approach in which horizontal elements became the basis for constructing diagrams. In fact, his invention of such planning was later called the Gantt chart.

Frederick Winslow Taylor (1856-1915) - American engineer, mathematician and physicist, his work is still used as prototypes of various modern tools, which includes the hierarchical structure of his work.

Henri Fayol (1841–1925) was a French engineer who founded the main classical project management model. He became quite famous due to the fact that he described the 5 main functions of management (management), which, in fact, became the basis of the theory of project management.

Theoretical aspects of research on project management are contained in the works of such Ukrainian and foreign scientists as, in particular, OO Kulinich, SD

Bushuyev, Yu.S. Грисюк, I.B. Kononenko, VI Максимова, В.Б. Силова, Є.К. Корноушенко, О.Г. Timinsky, VI Prangishvili, Dolores Sherwood Steiger, Yu.M. Tesla, Paula Martin, Karel Tate, L.A. Zade, Denise Colonna d'Istria. In their works, they covered the development of information technology in project management.

The reason for the relevance of an effective planning process is the need to increase human capacity to solve unique problems. Quite often companies implement several projects at the same time, which involve the same human or technical resources. Automating this process can increase the efficiency of the company as a whole. To date, there are many examples of efficiency by automating monotonous manual work.

Connection of work with scientific programs, plans, themes. The work was performed at the Department of Automated Information Processing and Control Systems of the National Technical University of Ukraine "Kyiv Polytechnic Institute. Igor Sikorsky "in the framework of the topic "Effective methods for solving problems of schedule theory" (№ DR 0117U000919).

The purpose of the study is to increase the efficiency of IT projects in the company, by creating an information system for resource planning of IT projects, which is based on methods that minimize the time of project implementation.

To achieve this goal it is necessary to perform the following **tasks**:

– perform an analytical review and conduct a detailed analysis of existing systems of resource planning, project management, management of constraints and risks within the project, methods of planning schedules for tasks within resource constraints;

– develop algorithms for solving the problem of supporting the work of the resource planning system of IT projects;

– investigate the effectiveness of the developed algorithms;

– develop software implementation of information system for resource planning of IT projects;

– perform an experimental study of the developed information system, methods of solving the problem and the effectiveness of the algorithms used.

The object of research is the process of planning IT project resources.

The subject of research - there are methods of automation of resource planning of IT projects.

The research methods used in this paper are based on methods for solving the problem of resource planning.

The scientific novelty of the obtained results is to build a method of resource planning, taking into account their limitations, risks and analysis; development and improvement of algorithms for solving the problem by performing experimental research.

Publications. Materials of the work are published in the collection of the Fifteenth International Scientific and Practical Conference MATHEMATICAL AND SIMULATION MODELING OF SYSTEMS (MODS 2020), as well as in the materials of the V All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Management and Technology" Information.

PROJECT MANAGEMENT, SCRUM, BACKLOG, SPRINT, RESOURCE, SCHEDULE THEORY, MINIMIZATION OF TASK TIME

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

ІС	інформаційна система
МОД	мова опису даних
ОПР	особа, яка приймає рішення
ОС	операційна система
ПК	персональний комп’ютер
СППР	систем підтримки прийняття рішень
ТР	теорія розкладів

ВСТУП

Управління проєктом [1] — це методологія організації, планування, керівництва, координації трудових, фінансових та матеріально-технічних ресурсів протягом проєктного циклу, спрямована на ефективне досягнення його цілей шляхом застосування сучасних методів, техніки та технології управління для досягнення певних результатів щодо складу та обсягу робіт, вартості, часу, якості та задоволення учасників проєкту.

До кожного проєкту визначаються мета, результати, склад та обсяги робіт. У процесі життя проєкту всі складові можуть зазнавати змін, вони можуть змінюватися або уточнюватися як у процесі розробки проєкту, так і у ході досягнення проміжних результатів.

Будь-яка функція управління складається з п'яти відносно самостійних видів управлінської діяльності: планування, організації, координації, активізації та контролю [1].

Функціями управління є управління якістю проєкту, часом, контрактами й постачаннями, комунікаціями в проєкті, ризиками, персоналом, різними видами ресурсів, а саме людськими, вартісними, технічними та іншими. Розглянемо деякі з них.

Управління людськими ресурсами в проєкті включають визначення потреби, чисельного і кваліфікаційного складу персоналу на всі періоди часу здійснення проєкту; пошук і відбір кандидатур, оформлення прийому на роботу і звільнення; планування і розподіл робітників по робочих місцях; організацію навчання і підвищення кваліфікації; установлення відповідальності; створення умов і робочої атмосфери для колективної роботи; попередження й вирішення конфліктів, що виникають; питання оплати та інше.

Окрім цього, в процесі роботи потрібно також враховувати і управління параметрами часу, що включають: визначення робіт і їхньої тривалості, строків початку і завершення проєкту, його частин, мінімізацію (оптимізацію) характеристик; прогнозування термінів завершення робіт, етапів і проєкту в цілому; прийняття

рішень щодо ліквідації небажаних тимчасових відхилень. Управління часом реалізується за допомогою процесів тимчасового аналізу проєкту і його частин, календарного планування робіт, контролю графіків виконання робіт, їх актуалізації та коригування. Під час реалізації проєкту саме часові ресурси і бувають порушеними через не ефективне планування людських ресурсів, що призводить до порушення термінів виконання проєкту та втрати фінансових ресурсів проєкту.

Одним із способів удосконалення управління проєктами розробки програмного забезпечення (ПЗ) є створення і використання системи підтримки прийняття рішень, що дозволяє виконувати складні операції, пов'язані зі складанням розкладу, а також за результатами аналізу поточного стану проєкту формувати вплив на реалізацію проєкту.

Незважаючи на велику різноманітність існуючих автоматизованих систем управління проєктами (Microsoft Project, Concerto, JIRA, Serena Team Track, Primavera Project Planner, Spider Project), безліч важливих процесів і завдань управління проєктами, до якого входить планування, не має необхідної комп'ютерної підтримки, або вона здійснюється недостатньо повно.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ПО УПРАВЛІННЮ ІТ-ПРОЄКТАМИ

1.1 Підходи до управління ІТ-проєктами

За час людської діяльності було зібрано чимало успішно реалізованих проєктів, в тому числі ІТ-проєктів. Починаючи з будівництва Пірамід у Гізі до першого польоту людини на Місяць, такі сміливі ідеї потребували злагодженої роботи величезної кількості людей.

Всі проєкти різні та кожен з них несе у цей світ щось своє. Немає ідеальної системи для управління проєктами, яка однаково підходить і може бути використана для кожного з типів проєктів. Необхідно розуміти, що немає системи, яка б підходила кожному керівнику того чи іншого проєкту. Але за час існування поняття «управління проєктом» було знайдено чимало ефективних підходів, методів і моделей, що можна використовувати для управління проєктами [2-5].

На сьогодні існують основні наступні методи управління проєктами:

– класичний проєктний менеджмент (waterfall);
– Scrum;
– Lean;
– Kanban;
– Six Sigma;
– PRINCE2.

Методи управління проєктом використовуються вже давно. Єгипетські піраміди і Велика Китайська стіна є реалізованими проєктми, де також автори вдавались до різних методів управління проєктом. На жаль, документальних записів як саме проходила реалізація та управління такими проєктами було втрачено, а сучасне проєктне управління значно відрізняється від знань минулих століть.

Найбільш ефективним напрямом реалізації проєкту є розбиття проєкту на фази або ж окремі завдання різних розмірів. Одним із найпростіших інструментів цього є чек-лист дій, що необхідно зробити для досягнення поставленої мети.

Далі розглянемо детальінше деякі методи управління проєктами.

Waterfall. Класичний метод управління проєктами, базується на так званому каскадному циклі або ж «водоспадному», при якому завдання рухаються послідовно по ітераціях, які нагадують потік [6-7].

Такий підхід доцільно використовувати для проєктів, які містять жорсткі обмеження стосовно послідовності виконання певних завдань та обмеження ресурсів. Наприклад, будівництво нового житлово-комунального комплексу – не можливо будувати поверхи будинків, якщо ще не було закладено фундамент.

Згідно з поетапністю цього підходу, виділяють 5 головних етапів управління проєктом за допомогою підходу Waterfall (хоча є можливість доповнювати додатковими етапами).

5 етапів Waterfall:

Етап 1. Ініціалізація. На цьому етапі визначаються вимоги до проєкту. Це виконують керівник проєкту та команда. Під час ініціалізації часто проводять так звані «Мозкові штурми», де швидше і легше можна зрозуміти, що має представляти із себе кінцевий продукт.

Етап 2. Планування. Під час етапу планування команда визначається, які завдання потрібно зробити задля досягнення поставленої мети. Проводиться уточнення та деталізація завдань проєкту, а також результати проєкту. На основі зібраної інформації команда створює календарний план, формує бюджет, проводить оцінку ризиків і можливих змін у ході роботи.

Етап 3. Розробка. Під час етапу розробки, основною характеристикою для технологічних проєктів, є те, що тут обирається конфігурація майбутнього проєкту та технічні методи його досягнення.

Етап 4. Реалізація та тестування. Протягом цього періоду виконується основна робота – написання коду. Контролюючи дотримання попередньо розробленого календарного плану формується наповнення проєкту, відбувається контроль за критеріями. У другій половині цього етапу проводять тестування продукту, де проходить порівняння та відповідність продукту вимогам замовника. Після тестування визначаються і коригуються недоліки продукту.

Етап 5. Моніторинг і завершення проєкту. На цьому етапі все залежить від типу проєкту, тому що цей етап може містити у собі передачу отриманих результатів реалізації проєкту замовнику або ж з доволі затяжного процесу взаємодії із замовниками щодо покращення результатів проєкту і поліпшенню рівня відповідності вимогам замовника.

Очевидно, що класичний метод управління проєктом строго прив'язаний до часової тривалості виконання завдань, яка попердньо була визначена на етапі планування, тоді в рамках цього методу для реалізації проєктів чудово підходять технології календарно-сітьового планування [8-11].

Переваги Waterfall:

– визначення потреб та вимог замовника на першому етапі проєкту;

– етап ініціалізації підвищує рівень стабільності по роботі з проєктом;

– чітке планування надає можливість впорядкувати виконання завдань проєкту;

– у етапи входить важлий етап з моніторингом та тестування отриманого продукту.

– керівник проєкту володіє повною інформацією про ресурси проєкту, навіть якщо їх оцінка була неточною, завжди враховано ризики порушення цих оцінок.

Недоліки Waterfall:

– основний недолік – будь-які зміни несуть за собою гострі порушення і відхилення від календарних планів, тобто негнучкість до змін під час реалізації проєкту на пізніх етапах.

Agile. Гнучкий ітеративно-інкрементальний метод управління проєктами, головний орієнтир якого направлений на динамічне формування вимог, можливість їх змін під час роботи та забезпечення дотримання даних вимог у результаті постійної взаємодії між самоорганізованими учасниками команди [12-17].

Далеко не всі проєкти можуть бути спроєктовані так, щоб їх можна було реалізувати за допомогою класичного методу управління проєктом. Згідно з даним підходом, проєкт не розділяється на послідовні етапи, він розділяється на невеликі підпроєкти, які потім будуть сформовані в готовий продукт.

Переваги Agile:

– гнучкість до змін та адаптивність;

– підлаштування під будь-які умови та процеси;

– можливість реалізації проєкту «шматками»;

– відслідковування виконання завдань за допомогою основних принципів підходу.

Недоліки Agile:

– необхідно адаптувати даний підхід під кожну команду виконавців окремо, а це непростий і доволі тривалий процес, який потребує окремих затрач часу на переналаштування.

1.2 Використання Scrum методології у рамках управління проєктом

Scrum розбиває проєкт на підпроєкти (частини основного проєкту), які замовник може використовувати фактично відразу. Потім ці частини розділяються за пріоритетом завдань (зазвичай це робить власник продукту). Головні «частини» першими виносять на реалізацію в спринт (ітерації в Scrum), що тривають від 2 до 4 тижнів.

Наприкінці спринта замовнику представляється робоча частина продукту – ті «шматки» проєкту, які вже можна використовувати у роботі. Після першого спринту команда виконавців приступає до наступного. Тривалість спринта завжди фіксована, але команда визначає її самостійно спираючись на складність завдань та доступну потужність команди, згідно з роботами [18-22].

Scrum складається з команд Scrum та пов'язаних з ними ролей, подій, артефактів та правил. Кожен компонент в рамках служить певній меті і є важливим для цього. Правила Scrum пов'язують між собою ролі, події та артефакти, керуючи стосунками та взаємодія між ними.

На сьогодні, Scrum є однією з найпопулярніших методологій розробки програмного забезпечення та підходів до управління проєктом. Відповідно до визначення, Scrum – це макет розробки, за допомогою якого люди мають можливість вирішувати проблеми, які було виявлено, при цьому успішно, продуктивно та ще й готуючи продукти високої якості та значущості (з точки зору клієнта – прим. автора) [19].

Це пояснюється тим, що в Scrum фактично неможливо знайти відповіді на всі поставлені проблеми та вказівки до дії у певних ситуаціях (наприклад, в офіційному описі Scrum лише зазначена наявність оцінки часу, що потрібен на виконання поставлених задач, але без уточнення, якого виду ця оцінка повинна бути.

Говорячи про підходи та методології Scrum, частіше за все мають на увазі саме гнучку методологію розробки програмного забезпечення, побудовану на основі правил і практик Scrum.

Авторами Scrum заявлені наступні особливості:

– легкий (англ. Lightweight);
– зрозумілий, доступний;
– складний в освоєнні (практично взаємовиключні параграфи).

У класичному Scrum існує 3 базових ролі:

– Product owner (власник продукту);
– Scrum master (скрам майстер);
– Development team (команда розробників).

Власник продукту (ВП) є ключовим «комутатором» між командою розробки та стороною замовника. Мета роботи ВП – максимально доступне збільшення цінності розроблюваного продукту та команди розробників будь-якими доступними методами у рамках роботи над проєктом.

Річ, яка допомагає ВП у його роботі над проєктом задля задоволення потреб замовника - є Беклог продукту (Product Backlog). Беклог продукту містить у собі усі необхідні для обов'язкового виконання робочі завдання (такі як користувацькі історії (User Story), баг (Bug), завдання (Task) та ін), кожне таке завдання містить у собі пріоритет (першочерговість виконання).

Scrum master (скрам майстер) є лідером команди розробників, на якому лежить уся відповідальність за використання методології. Мета роботи скрам майстра – допомогти команді розробників підвищити її ефективність за допомогою коректного налаштування робочого процесу, взаємодії між учасниками команди, навчання і мотивації команді для посилення командного духу.

Команда розробників (Development team) містить різних фахівців, які виконують роботу над продуктом, що виробляється.

В цілому, усіх разом виконавців (власник продукту, скрам майстер та команда розробників), які працюють з використанням Scrum, називають скрам-командою.

Якщо звернутись до документу «The Scrum Guide» (він є офіційним поясненням всіх особливостей Scrum від його авторів), команда розробників повинна володіти такими якостями і характеристиками:

– Самоорганізованість. Жоден зі скрам-команди не може вказувати команді розробників як їм реалізувати беклог продукту в працюючий продукт, який очікує замовник;

– Багатофункціональність, мультизадачність. Це саме проволодіння необхідними навичками лоя роботи в команді, продуктивності і реалізації працюючого продукту;

– Відповідальність. За виконану роботу (навіть якщо була допущена помилка) відповідає вся скрам-команда, а не індивідуально учасники команди.

Рекомендований розмір скрам-команди – 7-8 (вверх або вниз 2) людини. Згідно з класичними особливостями Scrum, команди великих розмірів потребують надто великих ресурсів на комунікації, але разом з цим команди маленьких розмірів підвищують ризики неуспішності проєкту (за рахунок достатньої кількості необхідних навичок та кваліфікації) і зменшують розмір виконаної роботи, який, в цілому, команда може виконати за одиницю часу реалізації проєкту[5-6].

Процес Scrum

Основою Scrum є його розділення на спринти (Sprint), протягом якого виконується робота над обраними завданнями для часткової реалізації продукту. Коли спринт завершується, повинно бути представлення виконаної роботи за спринт (нова робоча версія продукту або ж частина функціоналу). Спринт завжди обмежений у часі: зазвичай спринт триває 1-4 тижні і має однакову тривалість докінця реалізації проєкту.

На початку кожного спринта проводиться спринт планування (Sprint Planning), де проходить оцінка вмісту беклогу продукту і визначається (виділяється) спринт беклог (Sprint Backlog), який містить завдання різних видів, які мають бути реалізовані в поточному спринті і представлений результат їх виконання. У кожного спринта є своя певна мета (ціль), яка повинна слугувати мотивуючим фактором і може бути досягнена за допомогою виконання завдань з беклогу спринта.

У класичному Scrum, кожного дня проводять щоденна зустріч команди (Daily Scrum), на якому всі учасники команди відповідають на три основні питання «що було виконано вчора?», «що я буду робити на цей день?» «які

питання щодо завдань у мене є і можливо щось блокує мою роботу?». Мета таких зустрічей – визначити статус команди на даний момент і зрозуміти прогрес роботи над реалізацією спринта, завчасно виявлені проблеми чи запитання, які виникли у ході роботи, підготовка рішень по зміні тих чи інших моментів роботи, які є необхідними для досягнення цілей поточного спринта.

Коли спринт завершується, йде підготовка до спринт (перегляду Sprint Review) і спринт ретроспективи (Sprint Retrospective), основною метою яких є оцінка ефективності (або ж продуктивності) роботи скрам-команди в минулому спринті, провести прогноз відповідно до отриманих результатів на очікувану ефективність в наступному спринті, визначення незрозумілих завдань або запитань, оцінка ймовірності виконання виділених робіт по проєкту та інше.

Зображення роботи Scrum наведено на рисунку 2.2:

Рисунок 1.1 – Схематичне зображення процесу Scrum

Доволі часто зустрічається, що про Scrum говорять, що він не працює, або працює набагато гірше, ніж очікувалося напочатку. Необхідно зауважити, що найчастіше так відбувається з однієї з таких причин:

– Scrum застосовується невірно або неповністю.

Згідно з різними авторами по Scrum, досвід є основним джерелом достовірної інформації для його роботи і коректного використання. Необхідність наявності повноти та точності у використанні Scrum була прописана в «The Scrum Guide» і обумовлена нетиповою для класичних структур організацією процесу, де відсутні будь-які формальні фахівці, так як керівники чи лідери;

Одним з головних принципів Scrum є команди, що самоорганізуються, багатофункціональні команди. Згідно з дослідженнями соціологів, чисельність ініціатив співробітників, які здатні до самоорганізованості та мультизадачності. Цей показник не перевищує вього 15% від працездатного населення [23-25].

Звідки виходить, що лише маленька частина фахівців здатна ефективно працювати в скрам-командах без суттєвих змін у ролях скрам майстра і власника продукту, що, власне, і суперечить основній ідеї Scrum, і потенційно призводить до модифікованого використання Scrum, тобто:

– Scrum використовується для продукту, вимоги до якого суперечать головній ідеї скраму.

Scrum відноситься до сімейства Agile, так Scrum вітає зміни у вимогах в будь-який момент (беклог продукту може бути змінений в будь-який момент). Це приносить певні незручності використання Scrum у проєктах з фіксованим бюджетом (або часом). Згідно з головною ідеєю Scrum, що наперед неможливо передбачити та прорахувати всі зміни, тоді відсутній сенс наперед проводити планування всього проєкту, обмежившись тільки плануванням в даний момент часу, тобто враховувати тільки ті завдання, які повинні бути реалізовані в поточному спринті [26]. Звичайно, є і інші обмеження.

Переваги і недоліки

Scrum має доволі широкий список переваг. Він орієнтований на клієнта, адаптивний та гнучкий, надає клієнтові можливість вносити будь-які зміни до

вимог проєкту в будь-який момент часу (але немає ніяких гарантій, що саме ці нові зміни будуть виконані у проєкті). Можливість змінювати вимоги та умови поставлених завдань є привабливою для багатьох замовників.

Scrum доволі легко «присвоюється» для його вивчення, його можливість до економії часу за допомогою відходу від некритичних активностей. Scrum надає модливість отримати потенційно робочий продукт чи систему наприкінці кожного спринта.

Ідеологія скраму наголошує, що команду, яка є самоорганізованою, багатофункціональною та відповідальною, здатна вирішити необхідні завдання за рахунок мінімального числа будь-яких активностей. Це доцільно використовувати для малих компаній, а також стартапів, так як це надає можливість повністю відмовитись від необхідності спеціалізованого навчання співробітників та керівників.

Звичайно, у скрамі є і певні недоліки. Через свою простоту у використанні та мінімалістичність, скрам задає деяку кількість жорстких, некомфортних правил. Але це стає менш помітним, якщо всі у команді розуміють і приймають поняття «клієнтоорієнтованості», оскільки клієнтові байдуже на внутрішні правила та процеси скрам-команди, особливо якщо вони тим чи іншим чином обмежують замовника. Наприклад, у разі необхідності, за проханням клієнта спринт беклогу може бути змінений майжу повністю, не зважючи на повне протиріччя з правилами скраму.

Проблема є набагато більшою, аніж є насправді, оскільки скрам належить до сімейства Agile, в скрамі не прийнято, наприклад, планування внутрішніх комунікацій чи використання певних сценаріїв у момент наявності певних ризиків [25]. Таким чином, призводячи до формальної протидії, тобто до порушень усіх законів Scrum.

Також негативною точкою скраму є ще наголос на команду, яка самоорганізована, багатофункціональна. Хоча це надає можливість значно знизити витрати на роботу команди, але водночас це призводить до

підвищення вимог на відбір співробітників, їхню мотивацію. За певних умов ринку працевлаштування, створення повноцінної, ефективної скрам-команди може бути практично неможливим.

Для кращого розуміння суті управління ресурсами проведемо аналогію з управлінням фінансами результат представимо у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз управління ресурсами та фінансами

Фінанси	**Ресурси**
В певний момент виникає нестача коштів проєкту(наприклад, щоб виплатити зарплату – касовий розрив) – беремо кредит / займ	Потрібен конкретний ресурс на невеликий період часу (наприклад, щоб зробити термінову роботу без шкоди для основного бізнесу) - домовляємося «поширити» ресурс у сусідньому проєкті, де ресурс недозавантажений
Є надлишок грошей - кладемо їх в банк або інвестуємо в прибутковий проєкт.	Розробка "заблокована" тривалим узгодженням вимог замовником – віддаємо людей у тимчасове користування проєкту, де вони найпотрібніші. При цьому, витрати на цей період переносяться на сусідній проєкт, економимо бюджет свого проєкту
На ринку з'явилися дешеві гроші – перекредитуємо	На сусідньому проєкті звільнилися свої розробники, вартість яких нижче, ніж використовуваний на власному проєкті аутсорсний ресурс - робимо заміну

Загальна помічена закономірність – майже в кожному випадку управління ресурсами можна знайти аналогію з управління фінансами та навпаки. Однак, між грошима і ресурсами є дуже суттєва різниця, яка визначає специфіку цієї предметної області.

1.3 Задача планування ресурсів із застосуванням методології Scrum

Надійшло замовлення на реалізацію певного IT-проєкту. Було проведено дослідження (аналіз) даного проєкту та визначено необхідні ресурси та вимоги (умови) до його реалізації. Весь процес управління виконанням проєкту буде проходити за методологією Scrum, гнучкою методологією, що входить до сімейства Agile.

За результатами дослідження умов виконання проєкту було виявлено множину задач (завдань) до реалізації всього проєкту, які необхідно виконати, тобто формується беклог продукту. Кожній задачі присвоюється пріоритет щодо її реалізації (під пріоритетом розуміємо першочерговість реалізації задачі) та тривалість її виконання еталонним виконавцем (під еталонним виконавцем будемо розуміти виконавця, який працює з коефіцієнтом продуктивності 1).

Встановимо наступну класифікацію пріоритетів: високий, середній та низький (задачі з високим пріоритетом реалізуються першочергово, відповідно задачі з низьким пріоритетом реалізуються в останню чергу, можуть бути взагалі не реалізованими або такими, що відкладено до реалізації в наступних версіях продукту).

В Scrum для поступової реалізації проєкту використовують ітерації фіксованої довжини, що називаються спринтами, які зазвичай тривалістю 2-4 тижні. Щоб розпочати реалізацію у рамках спринта, з беклогу продукту виділяється беклог спринта, який містить у собі певну кількість оцінених задач. Беклог спринта формується за допомогою пріоритету задач: спершу всі задачі з високим пріоритетом, потім із середнім, потім з низьким, представлено на рисунку 1.2.

БЕКЛОГ ПРОДУКТУ		
Номер завдання	Пріоритет	Тривалість (години)
1	Високий	4
2	Високий	7
3	Низький	3
4	Високий	2
5	Середній	5
6	Середній	7
7	Високий	9
8	Низький	2
9	Середній	1
10	Низький	4
…		
80	Середній	8
Всього: 80 завдань		200 годин

БЕКЛОГ СПРИНТА		
Номер завдання	Пріоритет	Тривалість (години)
1	Високий	4
2	Високий	7
4	Високий	2
7	Високий	9
…		
Всього: 30 завдань		90 годин

Рисунок 1.2 – Схематичне зображення виділення з беклогу продукту беклогу спринта

Кількість завдань має бути такою, щоб вона не перевищувала значення продуктивності команди. Ця підмножна завдань стає беклогом першого спринта.

Над реалізацією даного IT-проєкту працює команда виконавців, кожен з яких має свій коефіцієнт продуктивності відповідно до посади. Склад, посади та кількість виконавців команди вказано у замовленні. Визначимо посади виконавців наступним чином: junior, middle та senior. Посада виконавця middle має коефіцієнт продуктивності 1, тобто виконавець на посаді middle є еталонним виконавцем. Виконавець на посаді junior та senior мають відповідні коефіцієнти продуктивності.

У даній постановці задачі розглядаємо випадок, коли фахівці є взаємозамінними і можуть працювати паралельно. Виконавці відрізняються коефіцієнтами продуктивності їх роботи, що визначається посадою, яку вони займають.

При формуванні беклогу наступного спринта беклог продукту переглядається, можуть змінюватися пріоритети завдань, змінюватися самі завдання та ділитися на дрібніші.

Отже, реалізація беклогу спринта триває до директивного терміну, тобто часу до якого повинні бути завершені всі роботи спринта на рисунку 1.3.

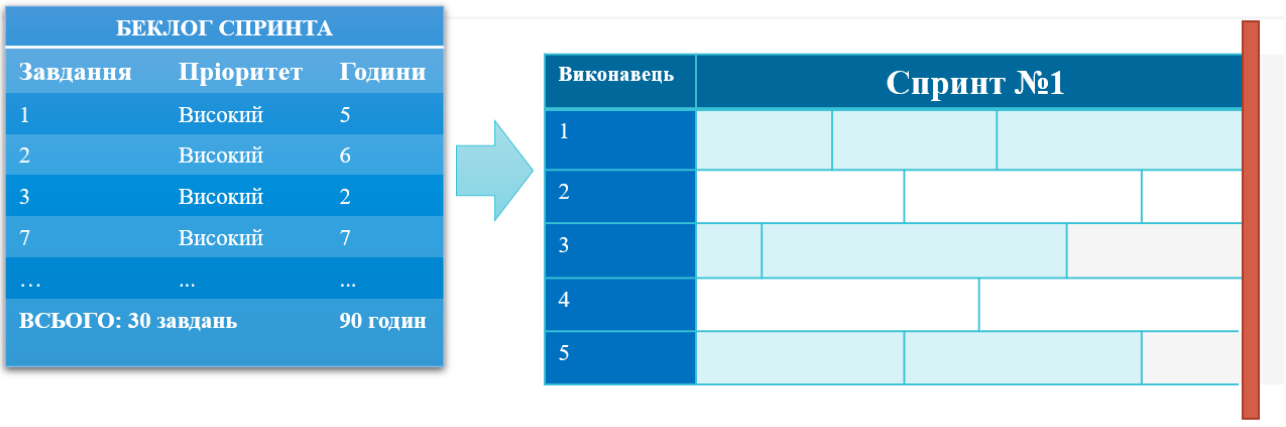

Рисунок 1.3 – Виділена підмножина завдань (беклог спринта) призначається на учасників команди

Оскільки метою поставленої задачі є мінімізація часу виконання проєкту та ефективне планування ресурсів (у даній задачі було обрано часові та людські ресурси), які є обмеженими, на допомогу визначенню властивостей та

особливостей даної задачі приходить теорія розкладів. Дана задача належить до одного з класів задач теорії розкладів та є NP-повною.

Одним з головних питань нового напрямку для задач планування була класифікація задач та точне визначення їх складності. Доволі сталою на сьогоднішній день класифікація задач ТР була запропонована Грехемом. Наповнені конструктивом та визначенностями, огляди по задачам ТР і їхньої складності було наведено у роботах Гері і Джонсона, Ленстра, Лоулераі, Танаєва, Брукера та інші [27].

Переважна більшість досліджених задач теорії розкладів є NP-складними. Незважаючи на це, практика вимагає рішення таких задач. Для цього існує кілька підходів. Першим з них є реалізація поліноміальних евристичних алгоритмів. Для багатьох евристичних підходів та алгоритмів вже відомі оцінки похибки отриманих результатів. Їх називають наближеними. Також є такі наближені алгоритми, які гарантують як відносну похибку отриманих результатів [28], так і абсолютну похибку результатів [29]. Деякі NP-складні задачі доводять, що існує певна апроксимаційна схема. З використанням даної схеми є можливість визначити наближене рішення задачі з відносною похибкою не більше будь-якого заданого значення $\varepsilon > 0$ за час, поліноміально залежний від $1 / \varepsilon$ і від розміру вхідних даних задачі, - цілком поліноміальна апроксимаційна схема (FPTAS).

Для завдань, що не мають апроксимаційної схеми, дуже важливо визначити так зване граничне значення ε, для якого є допустимим визначення ε-наближеного розв’язку за поліноміальний час - поліноміальна апроксимаційна схема (PTAS).

На даний момент широкого поширення мають мета евристичні алгоритми, які знаходять краще рішення, і це рішення є максимально наближеним до оптимального, за прийнятний час. Але недоліком таких та схожих алгоритмів є відсутність гарантії оптимальності отриманого рішення.

Неможливо визначити, наскільки розв’язок відрізняється від оптимального в найгіршому випадку.

Точним методам та підходам щодо рішення NP-складних задач також було приділено чимало досліджень та часу в роботах по ТР. Найбільш популярними стали методи скороченого перебору, або ж методи гілок і меж [22, 24]. Для скорочення перебору обчислюються нижні оцінки цільової функції (в разі її мінімізації) і застосовуються комбінаторні властивості задач. Для розв’язання задачі ТР широко використовується метод динамічного програмування [1, 6, 19].

Доволі часто виникає ситуація, що задачі теорії розкладів можуть бути поставлені як задачі цілочислового лінійного програмування. Для розв’язання таких задач присвячено чимало досліджень та робіт, наприклад, роботи [3, 4, 9, 19].

Станом на сьогодні, доволі часто використовують метод програмування в обмеженнях. Серед областей його ефективного використання є також і теорія розкладів [5].

Деякі складні задачі ТР можуть бути розв’язані за допомогою алгоритмів, які використовують елементи одночасно декількох методів. Їх називають "гібридні алгоритми" [8]. Даний напрямок, на наш погляд, є одним з перспективних.

1.4 Огляд існуючих рішень задачі планування ресурсів

На сьогодні, питання проєктування інформаційної системи планування ресурсів ІТ-проєктів в основі яких лежить задача теорії розкладів розглядало чимало вчених.

Так у роботі [30] авторами розглянуто головні переваги та слабкі місця використання інформаційних технологій. Проведено аналіз завдань та елементів системи управління проєктами саме зі сторони автоматизованої системи. Аналізуються ІС, які містять реалізації функцій календарного

планування і контролю якості виконання проєктів, було виділено їх плюси та мінуси використання їх у практиці. Проводиться детальний аналіз ефективності впровадження таких систем управління проєктами, які містять у собі планування комплексу робіт і безпосередній контроль їх реалізації. Надано детальний огляд методики «Total Cost Of Overship», яка надає можливість встановлювати оцінку ефективності витрат на проєкти.

У роботі [31] визначено сутність понять «проєкт», «план», «програма», а також пояснено різницю між ними. Виявлено заходи, через які здійснюється управління проєктами (проєктний менеджмент). Досліджено роль та значення інформаційних технологій в управлінні проєктами, а також виявлено основні переваги під час використання інформаційних систем управління проєктами (ІСУП). Виділено відомі інформаційні системи для управління проєктами, проведено огляд оцінки ефективності використання систем управління проєктами в Америці.

У роботі [32] було досліджено багатовимірне подання даних для управління ІТ-проєктами. У роботі обґрунтовано теоретичні положення, наведено методичні та практичні рекомендації, що дають змогу підвищити дієвість функціонування інформаційної системи. Наведено результати аналізу основних принципів та методів управління проєктами інформаційних процесів та обґрунтовано методологію формування корпоративної інформаційної системи управління проєктами. Розглянуто процес створення ІТ-проєкту, що дає змогу врахувати загальні стратегічні цілі розвитку, об’єднання та формування моделі для експлуатації системи підтримки прийняття рішень.

Робота [33] присвячена розробці моделі розподілу ресурсів ІТ-проєктів. Проведено аналіз існуючих економіко-математичних моделей, а також можливості їх застосування для прийняття рішень в системі управління персоналом ІТ-підприємства. Побудована динамічна оптимізаційна багатокритеріальна економіко-математична модель оптимального розподілу ІТ-проєктів по ІТ-командах. Запропонований метод знаходження

оптимального рішення з використанням розробленої економіко-математичної моделі.

Було розкрито принципи використання інформаційних технологій в управління проєктами саме підприємств у роботі [34] Автор розглянув особливості використання інформаційних технологій в управлінні проєктами підприємств. Обґрунтовано необхідність використання інформаційних технологій, досліджено механізми впровадження інформаційних технологій в управління проєктами.

У роботі [35] запропонували в системі управління проєктами використовувати інформаційні технології. Автори особливо наголошують на тому, що перспективність використання інформаційної технології управління ресурсами пояснюється широким упровадженням, високим рівнем фінансових запасів, доволі вузькою спеціалізацією та конфіндеційністю відомих рішень.

Управління ресурсами таких проєктів має ґрунтуватись на поєднанні процедурного і ресурсного методів та відрізняється від існуючих універсальністю використання, орієнтацією на застосування комплексу математичних моделей та методів, спрямованих на покращення режимів використання ІТ-проєктів, враховуючи головні чинники впливу на ефективність функціонування проєктів та планування ресурсів, а також на якість осблуговування.

Що ж стосується підходу саме до вирішення задачі планування, нище наведено теж декілька робіт (статті) для виділення основних проблем цих рішень або неможливості вирішення поставленої задачі.

Розроблені нові фундаментальні принципи управління та контролю стану ІТ-проєкту. Було опубліковано великий обсяг науково-популярних матріалів стосовно ІТ-галузі та проводиться різностороннє навчання [29]. Така ситуація у ІТ індустрії була внесена для значного покращення економічної ситуації країни та рівня життя у ній в цілому.

У роботі [36] наведено огляд обмежених ресурсами методів планування проєкту за різних цілей планування. Було показано, що вибір відповідного завдання планування - одне з головних рішень, яке потрібно прийняти на етапі планування проєкту, оскільки він визначає зовнішній вигляд та форму плану проєкту та ефективність використання ресурсів. Через притаманну складність планування проєктів із відновлюваними ресурсами за обмеженої доступності було використано багато швидких та простих евристичних методів, але не було представлено точних результатів роботи.

У роботі [37] оцінено можливості вирівнювання ресурсів з трьох програмних пакетів управління проєктами на двох реальних проблемах. Результати показують, що проєктна тривалість залежить від програмного забезпечення або використовуваного методу. Це можна підтвердити тим, що для проблем такого розміру, складності та цільової функції цей метод виконує мінімізацію панорамування. Спостереження підтвердило збільшення розробок виробництва з 41,11% до 167,79% ресурсу необмежений графік і що Primavera P6 перевершує MS Project і Open Workbench. Через великий діапазон результатів керівникам проєктів не слід «довіряти» першим результатам, які вони отримують, а спробувати інші правила або навіть програмне забезпечення, якщо це можливо.

У роботі [38] в порядку для оцінки кожного потенційного рішення, тобто представлених пріоритетів задач, використовується паралельна схема перетворення цієї частинки на можливий графік відповідно до переваги та обмеження ресурсів. Обчислювальний аналіз показує, що даний підхід має можливість пошуку глобальних оптимумів (тобто оптимальних графіків з мінімальною тривалістю проєкту) і є більш ефективним, ніж підхід евристики завдяки до таких його функцій, як односторонній механізм обміну досвідом під час пошуку рішення. Підхід забезпечує ефективну та просту у реалізації альтернативу проаналізувати та досягти ефективного результату.

Висновки до розділу

У першому розділі даної дипломної роботи було розглянуто різні підходи до управління проєктами, наведено конкретні моделі, а також виділено їхні переваги та недоліки. Було детально розглянуто підхід до управління IT-проєктом – Scrum. Наведено повний опис бізнес-процесів, які відбуваються під час проєктного управління.

Наведено розгорнуту постановку задачі планування ресурсів у рамках управління проєкту з використанням підходу Scrum. Визначено основні етапи під час планування ресурсів. Здійснено детальний аналіз існуючих рішень, які вирішують схожу задачу планування. Для кожного існуючого рішення був наведений опис, з основною його метою та результатами, яких зміг досягти автор під час дослідження певної задачі.

2 РОЗВ'ЯЗАННЯ ЗАДАЧІ ПЛАНУВАННЯ РЕСУРСІВ ІТ-ПРОЄКТІВ

2.1 Математична постановка задачі

Дано множину завдань проєкту $J = \{1,..., j,..., n\}$ та множину виконавців (учасників команди) $M = \{1,..., i,..., m\}$.

У системі всі завдання надходять одночасно в нульовий момент часу та усі завдання мають спільний директивний термін d .

Процес обслуговування кожного завдання проходить без переривання до завершення виконання.

Вважається, що кожен учасник команди має свій коефіцієнт продуктивності виконання завдань – $k_i > 0$, а кожне завдання має свій пріоритет – $p_j > 0$, та початкову тривалість виконання $t_j > 0$ $j = \overline{1,n}$ (початкову – тобто еталонну тривалість, за яку може бути виконане завдання еталонним виконавцем, коефіцієнт продуктивності якого $k_i = 1$), тоді тривалість виконання j-ого завдання i-им учасником команди, з урахуванням коефіцієнту продуктивності учасника становить:

$$t_{ij} = t_j * k_i . \qquad (2.1)$$

Нехай C_j – момент завершення j-ого завдання, $C_{\max} = \max_j\{C_j\}$ – максимальний момент завершення завдань (момент завершення останнього завдання), C_i – момент завершення виконання завдань i-им виконавцем, $Z_j = \max_j\{0, C_j - d\}$, – запізнення завдання; $E_j = \max_j\{0, d - C_j\}$ – випередження завдання; $E = \sum_{j=1}^{n} E_j$ – сумарне запізнення; $Z = \sum_{j=1}^{n} Z_j$ – сумарне випередження.

Критерій 1. Необхідно скласти розклад виконання завдань, при якому досягає мінімуму максимальний час завершення завдань:

$$C_{\max} \to \min . \qquad (2.2)$$

Критерій 2. Необхідно скласти розклад виконання завдань, при якому буде рівномірно розподілено завдання між виконавцями з урахуванням їх коефіцієнту продуктивності:

$$\max_j \{E_j, Z_j\} \to \min . \qquad (2.3)$$

2.2 Дослідження властивостей задачі

Вважається, що процес обслуговування кожного завдання у даній задачі проходить без зупинок до завершення виконання завдань.

Спочатку визначається питання про можливість одночасного виконання завдання декількома виконавцями. У цьому випадку - це неможливо, тоді задача зводиться до розподілення n завдань на m виконавців, які не перетинаються.

Завдання може бути призначене тільки на одного виконавця у певний момент часу і виконуватись ним до завершення виконання завдання.

Для систем, де кожне завдання виконується одним виконавцем та тривалості завдань змінюються від одного виконавця до іншого у довільному порядку, залишаються невідомими загальні результати виконання завдань. Але, якщо виконавців можна впорядкувати за швидкість виконання завдань (для даної задачі – за коефіцієнтом їх продуктивності), а цей порядок є однаковим для всіх завдань, то наявний алгоритм, який мінімізує середню тривалість виконання завдання.

Тоді, кожен виконавець має коефіцієнт продуктивності виконання завдань – $k_i > 0$, а кожне завдання має свій пріоритет – $p_j > 0$, та початкову тривалість виконання $t_j > 0 \;\; j = \overline{1,n}$. Початкову – це еталонна тривалість

виконання завдання, за яку може бути виконане завдання еталонним виконавцем, у якого у свою чергу коефіцієнт продуктивності дорівнює одиниці ($k_i = 1$).

Оскільки було внесено ще певні позначення, розглянемо їх детальніше нище:

– C_j – момент завершення j-ого завдання;

– $C_{\max} = \max_j\{C_j\}$ – момент завершення останнього завдання (максимальний момент завершення завдань);

– C_i - момент завершення виконання завдань i-им виконавцем.

Вважається, що кожне завдання має директивний термін (або плановий) - d. Значення директивного терміну являє собою момент, коли повинне бути виконане завдання і не пізніше заданого терміну.

На початку аналізу даної задачі постало питання визначити нижню границю загальної тривалості виконання завдань (нижня границя вищенаведених критеріїв оптимальності):

$$C^* = \frac{T}{K}, \tag{2.4}$$

де $T = \sum_{j=1}^{n} t_j$ - сумарний об’єм завдань у системі, $K = \sum_{i=1}^{m} k_i$ - сумарний коефіцієнт продуктивності.

Звідки виходить, що розклад, де усі учасники команди завершують виконання завдань в момент часу C^* є оптимальним за двома вищенаведеними критеріями.

Згідно з визначенням директивного терміну, далі представимо моменти запізнення та випередження j-ого завдання для формування критеріїв, а також їхні сумарні значення:

– $Z_j = \max_j\{0, C_j - d\}$, – запізнення j-ого завдання;

– $E_j = \max_j\{0, d - C_j\}$ − випередження j-ого завдання;

– $E = \sum_{j=1}^{n} E_j$ – сумарне запізнення завдань;

– $Z = \sum_{j=1}^{n} Z_j$ – сумарне випередження завдань.

2.3 Алгоритми розв'язання задачі

Для розв'язання задачі теорії розкладів, що відноситься до класу NP та за наявності декількох різних критеріїв оптимальності існує безліч різних алгоритмів розв'язання, кожен з яких має свої переваги та свої недоліки. Але для задач теорії розкладів доцільно використовувати евристичні алгоритми розв'язання. Не гарантовано, що оптимальне рішення буде знайдено, проте ці алгоритми значно зменшують час розв'язання таких задач, що забезпечує швидке отримання результату.

Тому для розв'язання задачі, описаної вище, у роботі обрано алгоритм локального пошуку [40] та жадібний алгоритм [41].

Жадібний алгоритм

Жадібний алгоритм простий для розуміння і реалізації, працює досить швидко, відомо багато задач різної складності, які можна розв'язати за допомогою жадібного алгоритму. Алгоритм проводить обробку завдань в довільному порядку. Кожне завдання він намагається призначити на першого виконавця, куди вона може бути призначена у відповідності до його продуктивності. Нище наведено схему роботи жадібного алгоритму.

Схема жадібного алгоритму

Крок 1. Відсортувати масив завдань у порядку спадання тривалості виконання завдань $t_1 > t_2 > ... > t_j > ... > t_n$.

Крок 2. Визначити i-ого виконавця, де $i = 1,...,m$, у якого момент завершення виконання завдань $C_i = C_{\min}$ і призначити на нього j-те завдання з найбільшою тривалістю t_1.

Крок 3. Повторити **Крок 2**, допоки n завдань з множини $J = \{1,..., j,..., n\}$ не буде розподілено на m виконавців.

Крок 4. Отримали розклад G.

Алгоритм локального пошуку

Натомість алгоритм локального пошуку зарекомендував себе як один з найкращих евристичних алгоритмів розв'язання задач, точні вирішення яких потребують експоненційних затрат часу.

Алгоритм локального пошуку є мета-евристичним алгоритмом, який базується на роботі з локальних пошуком у певному околі, щоб попередити його від попадання в безвихідь в локальних оптимумах, не дозволяючи ті переміщення, які призводять до повернення до попередніх знайдених рішень і повторення циклічної роботи алгоритму. Алгоритм локального пошуку бере початок з кінцевого рішення. На кожній ітерації генерується певна околиця розв'язків, і найкраще рішення з цієї околиці стає новим рішенням. Вибір найкращого доступного розв'язку в околиці проходить так, що він не бере жодного з недопущених атрибутів. Найкраще допустиме рішення оновлюється у тому випадку, якщо нове рішення краще і належить до допустимих. Процес не завершується, допоки не виконається одна з двох умов зупинки роботи алгоритму: максимальне число виконуваних ітерацій було досягнуто і максимально допустиме число ітерацій, коли поточне рішення не змінювалось.

Локальний пошук – це пошук, який виконується алгоритмами локального пошуку, у яких пошук відбувається тільки за рахунок поточного стану рішення, а раніше проведені результати не враховуються і не запам'ятовуються. Основною ціллю пошуку є не лише знаходження кращого шляху до цільової точки, а також оптимізація цільової функції задачі, тому, власне, задачі, що вирішуються подібними або схожими алгоритмами, називають задачами оптимізації.

Цей алгоритм зарекомендував себе як один з найкращих евристичних алгоритмів розв'язання задач, точні вирішення яких потребують експоненційних затрат часу.

Першим етапом побудови плану виконання завдань є побудова початкового розкладу за допомогою жадібного алгоритму.

Схема алгоритму локального пошуку:

Крок 1. Відсортувати масив завдань у порядку спадання тривалості виконання завдань $t_1 > t_2 > ... > t_j > ... > t_n$.

Крок 2. Визначити i-ого виконавця, $i = \overline{1,m}$, у якого момент завершення виконання завдань $C_i = C_{\min}$ і призначити на нього j-те завдання з найбільшою тривалістю t_1.

Крок 3. Повторити **Крок 2**, допоки n завдань з множини $J = \{1,..., j,..., n\}$ не буде розподілено на m виконавців.

Крок 4. Отримано розклад G.

Крок 5. З розкладу G визначити значення C_i між усіма виконавцями. З них обрати $C_{\max}$ та $C_{\min}$.

Крок 6. Повторити **Крок 1**, **Крок 2** та **Крок 3** для обраних виконавців з моментами завершення виконання завдань $C_{\max}$ та $C_{\min}$ для побудови розкладу G'.

Крок 7. Внести отриманий розклад G' до загального початкового розкладу G на місце виконавців зі значеннями $C_{\max}$ та $C_{\min}$ – отримали розклад ітерації 1 - G_1.

Крок 8. Для розкладу G_1 повторити крок 5, крок 6 та крок 7. Отримаємо розклад G_2.

Крок 9. Повторити Крок 8, допоки не буде достягнуто умову припинення пошуку: коли отримали найкращий розклад G_q за всі попередні на ітерації $q, q = 1,2,...,l$, а наступні $l, l = 1,..., f$ ітерацій з відповідними

розкладами G_l дали гірший (або такий же) результат, ніж на ітерації G_q. Значення *l* дано напочатку роботи алгоритму.

Досліджуючи схеми та властивості цих алгоритмів, очікуємо, що алгоритм локального пошуку надасть можливість отримати більш точний розв'язок, що більше відповідає встановленим критеріям, швидше ніж жадібний алгоритм.

2.4 Приклад розв’язання задачі

Розв’яжемо дану задачу (критерій 1) жадібним алгоритмом та представимо поетапно, що відбувається на кожному кроці.

Початкові дані наведемо в таблиці 2.1.

Таблиця 2.1 – Вхідні дані

Назва	**Значення**
Множина значень тривалості завдань	1, 1, 1, 2, 2, 2, 3, 4, 5, 5, 6, 6, 6, 7, 8, 9, 10
Множина коефіцієнтів продуктивності виконавців	1, 1, 2, 3

Множину завдань необхідно відсортувати за спаданням тривалості виконання завданння, а множину коефіцієнтів продуктивності виконавців можна залишити без змін. Визначаємо початковий момент завершення виконання завдань для кожного виконавця в таблиці 2.2

Таблиці 2.2 – Моменти завершення виконання завдань по кожному виконавцю

Виконавець (№)	**Момент завершення виконання завдань**
1	0
2	0
3	0
4	0

Eman 1. Візьмемо завдання 1 з тривалістю виконання 10 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, що наведено у таблиці 2.3.

Таблиця 2.3 – Визначення моменту завершення виконання завдання №1 для кожного виконавця

Виконавець (№)	**Момент завершення виконання завдання №1**
1	10
2	10
3	20
4	30

Далі визначаємо виконавця, у якого момент завершення виконання завдання №1 - найменший. Тобто завдання можна надати для виконавця №1 або виконавця №2, що зображено на рисунку 2.1

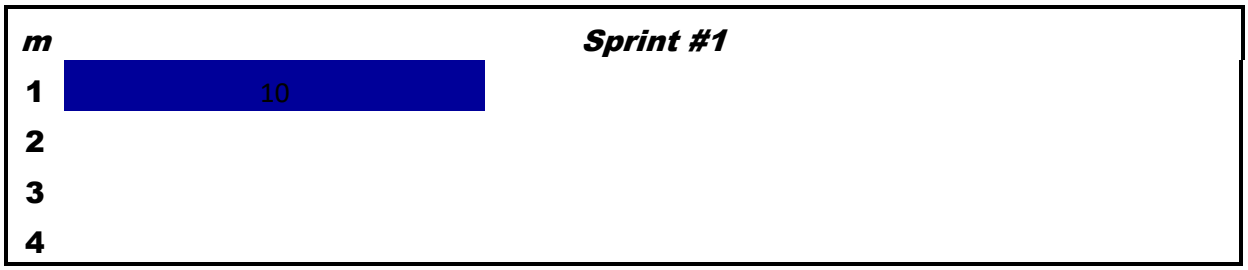

Рисунок 2.1 – Призначення завдання на виконавця №1

Збережемо останні зміни до таблиці з визначенням моменту завершення виконання завдання по кожному виконавцю, де надалі будемо фіксувати призначені завдання по кожному виконавцю, таблиця 2.4.

Таблиця 2.4 – Моменти завершення виконання завдань по кожному виконавцю

Виконавець (*m*)	**Моменти завершення виконання завдань по кожному виконавцю (C_i)**	**Етап 1**
1	10	1(10)
2	0	-
3	0	-
4	0	-

Примітка: *Надалі для зручності заповнення таблиці, будемо використовувати позначення m, C_i та нумерацію етапів (1, 2, 3 і т.д.) у шапці таблиці.*

Етап 2.

Візьмемо завдання 2 з тривалістю виконання 9 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 1. Результати наведено у таблиці 2.5.

Таблиця 2.5 – Етап 2: визначення моменту завершення виконання завдання №2 для кожного виконавця

Виконавець (№)	**Момент завершення виконання завдання №2**
1	19
2	9
3	18
4	27

Далі визначаємо виконавця, у якого момент завершення виконання завдання №2 - найменший. На етапі 2 завдання можна призначити на виконавця №2, що зображено на рисунку 2.2.

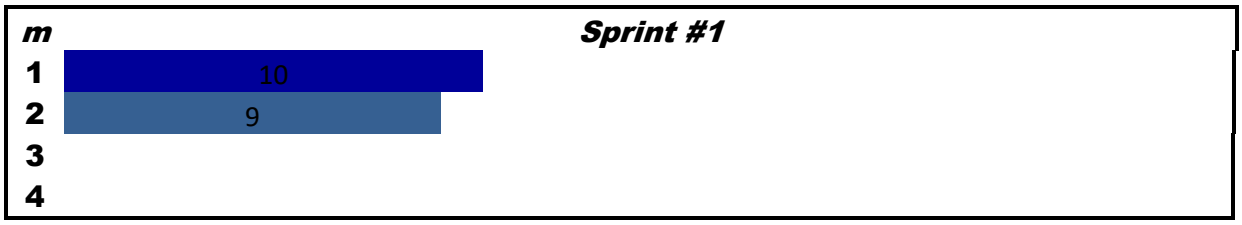

Рисунок 2.2 – Призначення завдання на виконавця №2

Внесемо отримані результати до таблиці 2.6.

Таблиця 2.6 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет.2**
1	10	1(10)	-
2	9	-	2(9)
3	0	-	-
4	0	-	-

Етап 3.

Візьмемо завдання 3 з тривалістю виконання 8 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 2. Результати наведено у таблиці 2.7.

Таблиця 2.7 – Етап 3: визначення моменту завершення виконання завдання №3 для кожного виконавця

Виконавець (№)	**Моменти завершення виконання завдання №3**
1	18
2	17
3	16
4	24

Далі визначаємо виконавця, у якого момент завершення виконання завдання №3 - найменший. На етапі 3 завдання можна призначити на виконавця №3, що зображено на рисунку 2.3.

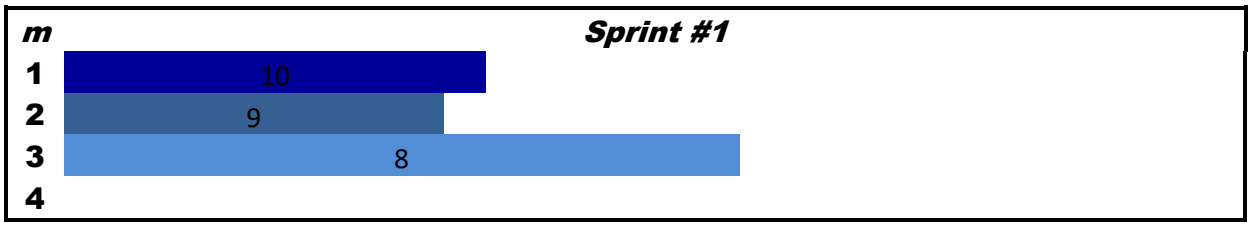

Рисунок 2.3 – Призначення завдання на виконавця №3

Внесемо отримані результати до таблиці 2.8.

Таблиця 2.8 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет.2**	**Ет. 3**
1	10	1(10)	-	-
2	9	-	2(9)	-
3	16	-	-	3(8)
4	0	-	-	

Етап 4.

Візьмемо завдання 4 з тривалістю виконання 7 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 3. Результати наведено у таблиці 2.9.

Таблиця 2.9 – Етап 4: визначення моменту завершення виконання завдання №4 для кожного виконавця

Виконавець (№)	**Моменти завершення виконання завдання №4**
1	17
2	16
3	30
4	21

Визначаємо виконавця, у якого момент завершення виконання завдань - найменший. На етапі 4 завдання можна призначити на виконавця №2, що зображено на рисунку 2.4.

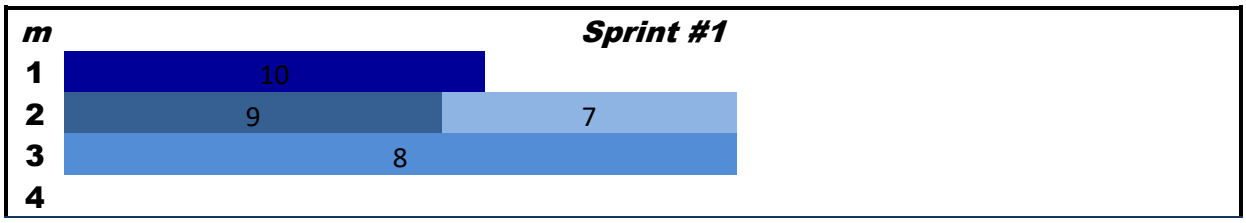

Рисунок 2.4 – Призначення завдання на виконавця №2

Внесемо отримані результати до таблиці 2.10.

Таблиця 2.10 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет.2**	**Ет. 3**	**Ет.4**
1	10	1(10)	-	-	-
2	16	-	2(9)	-	4(7)
3	16	-	-	3(8)	-
4	0	-	-		-

Етап 5.

Візьмемо завдання 5 з тривалістю виконання 6 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 4. Результати наведено у таблиці 2.11.

Таблиця 2.11 – Етап 5: визначення моменту завершення виконання завдання №5 для кожного виконавця

Виконавець (№)	**Моменти завершення виконання завдання №5**
1	16
2	22
3	28
4	18

Визначаємо виконавця, у якого момент завершення виконання завдань - найменший. На етапі 5 завдання можна призначити на виконавця №1, що зображено на рисунку 2.5.

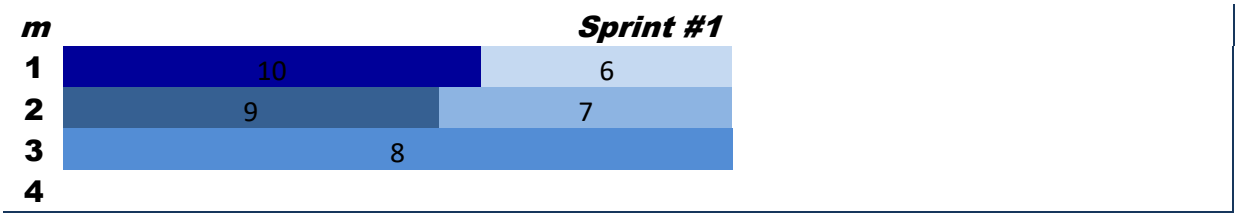

Рисунок 2.5 – Призначення завдання на виконавця №1

Внесемо отримані результати до таблиці 2.12.

Таблиця 2.12 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет.2**	**Ет. 3**	**Ет.4**	**Ет.5**
1	16	1(10)	-	-	-	5(6)
2	16	-	2(9)	-	4(7)	-
3	16	-	-	3(8)	-	-
4	0	-	-		-	-

Етап 6.

Візьмемо наступне завдання 6 з тривалістю виконання 6 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 5. Результати наведено у таблиці 2.13.

Таблиця 2.13 – Етап 6: визначення моменту завершення виконання завдання №6 для кожного виконавця

Виконавець (№)	**Моменти завершення виконання завдання №6**
1	22
2	22
3	28
4	18

Визначаємо виконавця, у якого момент завершення виконання завдань - найменший. На етапі 6 завдання можна призначити на виконавця №4, що зображено на рисунку 2.6.

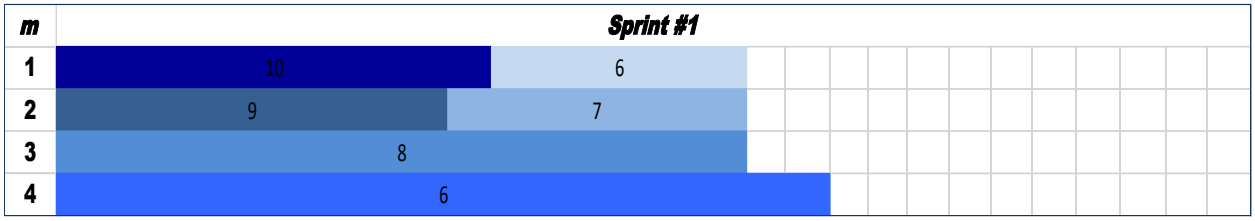

Рисунок 2.6 – Призначення завдання на виконавця №4

Внесемо отримані результати до таблиці 2.14.

Таблиця 2.14 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет.2**	**Ет. 3**	**Ет.4**	**Ет.5**	**Ет.6**
1	16	1(10)	-	-	-	5(6)	-
2	16	-	2(9)	-	4(7)	-	-
3	16	-	-	3(8)	-	-	-
4	18	-	-		-	-	6(6)

**Примітка: повторюємо аналогічні дії на кожному етапі протягом наступних 10-ти етапів. Далі розглянемо детально ще останні два етапи.*

Етап 16.

Візьмемо завдання 16 з тривалістю виконання 1 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 15. Результати наведено у таблиці 2.15.

Таблиця 2.15 – Етап 16: визначення моменту завершення виконання завдання №16 для кожного виконавця

Виконавець (№)	**Моменти завершення виконання завдання №16**
1	28
2	29
3	28
4	27

Визначаємо виконавця, у якого момент завершення виконання завдань - найменший. На етапі 16 завдання можна призначити на виконавця №4, що зображено на рисунку 2.7.

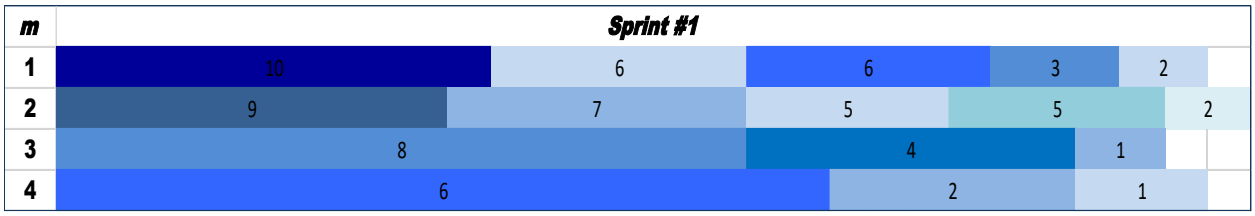

Рисунок 2.7 – Призначення завдання на виконавця №4

Внесемо отримані результати до таблиці 2.16.

Таблиця 2.16 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет.2**	**Ет. 3**	**Ет. 4**	**Ет. 5**	…	**Ет. 12**	**Ет. 13**	**Ет. 14**	**Ет. 15**	**Ет. 16**
1	27	1(10)	-	-	-	5(6)	…	-	13(2)	-	-	-
2	28	-	2(9)	-	4(7)	-	…	-	-	14(2)	-	-
3	26	-	-	3(8)	-	-	…	-	-	-	15(1)	-
4	27	-	-		-	-	…	12(2)	-	-	-	16(1)

Етап 17.

Візьмемо завдання 17 з тривалістю виконання 1 та обчислимо момент завершення виконання цього завдання для кожного з виконавців, враховуючи результати етапу 16. Результати наведено у таблиці 2.17.

Таблиця 2.17 – Визначення завантаженості машин

Виконавець (№)	**Моменти завершення виконання завдання №17**
1	28
2	29
3	28
4	30

Визначаємо виконавця, у якого момент завершення виконання завдань - найменший. На етапі 17 завдання можна призначити на виконавця №3, що зображено на рисунку 2.8.

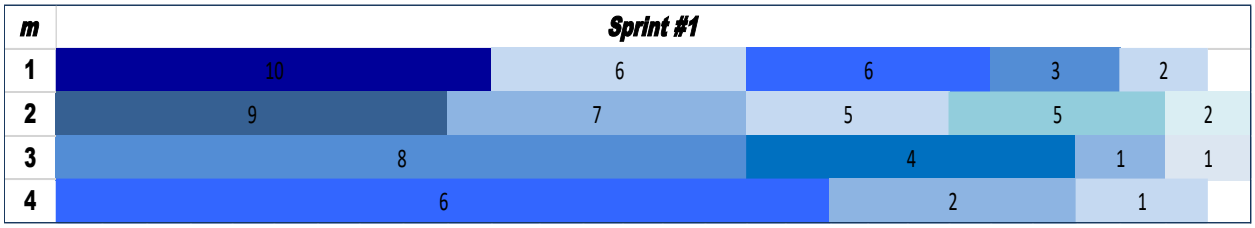

Рисунок 2.8 – Призначення останнього завдання на виконавця №3

Внесемо отримані результати до таблиці 2.18.

Таблиця 2.18 – Моменти завершення виконання завдань по кожному виконавцю

m	C_i	**Ет. 1**	**Ет. 2**	**Ет. 3**	**Ет. 4**	**Ет. 5**	…	**Ет. 12**	**Ет. 13**	**Ет. 14**	**Ет. 15**	**Ет. 17**
1	27	1(10)	-	-	-	5(6)	…	13(2)	-	-	-	-
2	28	-	2(9)	-	4(7)	-	…	-	14(2)	-	-	-
3	28	-	-	3(8)	-	-	…	-	-	15(1)	-	17(1)
4	27	-	-		-	-	…	-	-	-	16(1)	-

Отримали результати роботи жадібного алгоритму, де значення цільової функції – 28 (для виконавців №2 та №3).

Висновки до розділу

У даному розділі магістерської дисертації було представлено математичну постановку задачі, побудовано математичну модель, визначено критерії оптимальності задачі.

Досліджено властивості даної задачі, особливості завдань та виконавців, які знаходяться у системі. Обгрунтовано, які алгоритми розв’язання було обрано для розв’язання поставленої задачі, їхні переваги та недоліки, представлено схеми роботи алгоритмів.

Наведено деталізований приклад роботи жадібного алгоритму з таблицями, де записуються дані після виконання кожного етапу. Наведено результат роботи прикладу, значення цільової функції, яке отримали напикінці розрахунків.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис програмного забезпечення

У ході роботи над магістерською дисертацією була обрана доволі складна задача саме з точки зору реалізації її розв’язання, оскільки не один автор зустрічався з проблемою обрати найбільш ефективне середовище та технології для розробки декількох алгоритмів, які передбачають збереження всіх даних до бази, а також створення класифікатора задач, який потребує високого рівня швидкодії зі сторони обраних засобів для його реалізації. Нище наведено вимоги до програмного забезпечення та опис обраних технологій. У додатку А (плакат 3) зображено UML-діаграму варіантів використання.

3.1.1. Вимоги до програмного забезпечення

У таблиці 3.1 наведемо список функціональних вимог до ПЗ.

Таблиця 3.1 – Функціональні вимоги до програмного забезпечення

№	Назва	Пріоритет
1	Система надає можливість вводити вхідні дані задачі: – кількість завдань; – тип команди; – тип спринта.	Високий
2	Система надає можливість генерування задач	Високий
3	Система надає можливість вводити дані для генерування задач: – кількість задач; – кількість завдань; – тип команди; – тип спринта.	Високий
4	Система надає можливість показати отриманий розклад (результат)	Середній
5	Система надає можливість проводити експерименти	Високий

Продовження таблиці 3.1

№	Назва	Пріоритет
6	Система надає можливість відображати результати експериментів	Високий
7	Система надає можливість розв’язати задачу двома алгоритмами (алгоритм локального пошуку та жадібний алгоритм)	Високий
8	Система надає можливість проводити експерименти для двох алгоритмів	Високий

У цій системі кожна з наведених функцій реалізована за допомогою відповідного модуля, кожен з цих модулів містить можливість виконання певного переліку операцій, таких як редагування інформації, пошук даних та збереження результатів пошуку, формування звітів.

Функціонально-логічна структура програмного забезпечення, що розробляється у рамках магістерської дисертації наведено на рисунку 3.1.

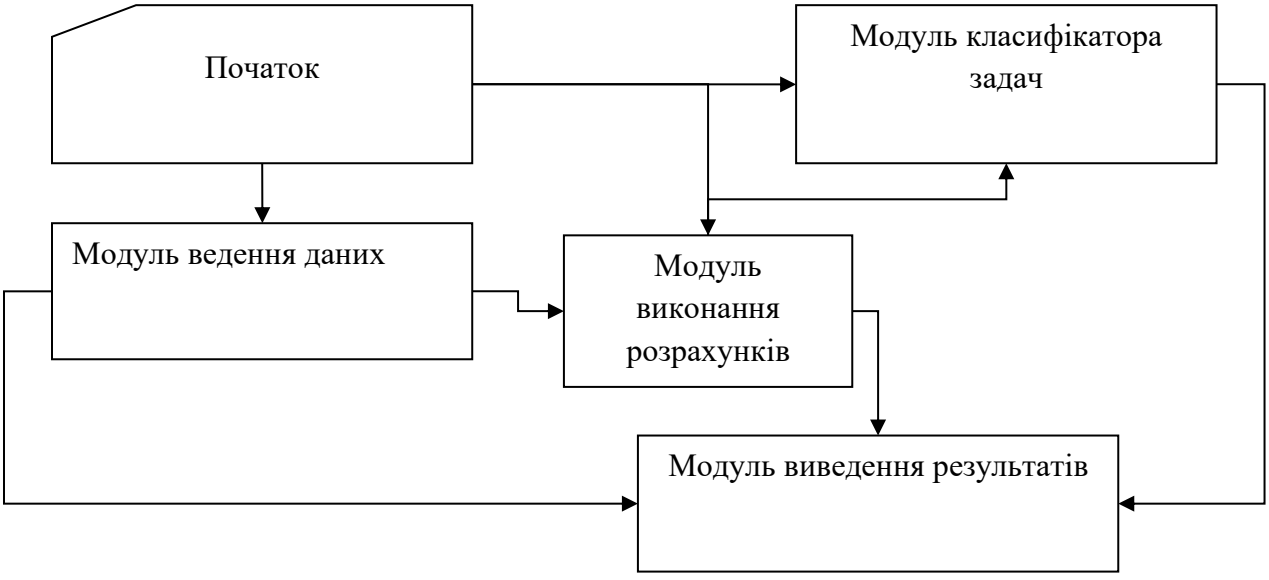

Рисунок 3.1 – Функціонально-логічна структура програмного забезпечення

Інформаційна система планування ресурсів ІТ-проєктів успішно пройшла випробування і готова до експлуатації.

Інформаційна система планування ресурсів IT-проєктів дозволяє підвищити оперативність в роботі; забезпечити простоту і зручність розрахунків; зменшити кількість помилок при планування ресурсів IT-проєктів відносно часових обмежень; підвищити якість планування; зменшити трудові витрати на обробку інформації.

Дана програма може використовуватись менеджерами та індивідуальними користувачами для автоматизованого планування ресурсів IT-проєктів, що скоротить час розрахунків та дозволить спостерігати за зміною значень по кожному окремому параметру проєкту (ресурсу).

Впровадження сучасних інформаційних технологій потребує високого рівня забезпечення фінансової частини. Функціонування компаній у ринковому середовищі вимагає як мінімум аналізу економічних наслідків, а краще – оцінки економічної ефективності того чи іншого кроку перетворення системи.

Впровадження системи залежить від багатьох факторів і є нетривіальним практичним завданням. При впровадженні використовується багатоетапна процедура, на кожному з яких здійснюється тісна взаємодія фахівців виконавця і замовника, контроль виконання пропонованих вимог і навчання персоналу.

Слід виділити три етапи впровадження інформаційної системи планування ресурсів IT-проєктів:

Дослідження. Спочатку проводиться дослідження предметної області і бізнес процесів.

Доопрацювання системи. Програмісти налаштовують або дороблюють необхідну частину системи.

Запуск системи. Початок використання системи у режимі реального часу, що включає в себе процеси навчання персоналу.

На етапі дослідження бізнес процесів відводиться певний час. Тут необхідно максимально точно описати, які процеси необхідно поліпшити.

Зазвичай, функціональність інформаційної системи буває в рази ширшо, ніж реальні процеси, що виконуються. На цьому етапі необхідно обрати як присутність різних функцій відобразиться на вартості системи, на часі виконання та впровадженні, і найголовніше, чи відповідатиме запропонована функціональність цілям компанії.

Очевидно, надзвичайно важливим моментом є і те, щоб результати дослідження були надані в якості окремого документа, де відповідно до вимог компанії повинен бути детально описаний процес діяльності.

Після стадії дослідження необхідно точно надати відповідь щодо вартості та термінів реалізації, а також запуску інформаційної системи.

Під час доопрацювання системи доволі влучно контролювати процес реалізації необхідних частин в ІС [43]. Важливо, щоб зі сторони Замовника виступала людина, яка прекрасно розуміється на меті, завдання з компанії та її бізнес процесами.

На фазі запуску системи необхідно переключити бізнес-процеси компанії на використання реалізованої системи.

Розроблена система, як правило, не починає працювати відразу. Потрібно проаналізувати наскільки впровадження було успішно, чи досягнуто основні цілі реалізації [44].

Впровадження вважається успішним, тільки тоді, коли система надає можливість отримувати вигоду, а саме покращує процес роботи різних відділів компанії, надає можливість виконувати роботу в рази швидше, підвищує якість роботи всіх учасників та різних процесів. Важливо постійно проводити аналіз показників роботи системи, а також ступінь зацікавленості співробіьників компанії у використанні саме цієї системи та місце цього використання. Процес впровадження ІС займає як правило, від кількох місяців до року. Під час цього етапу важливо фокусуватися на раніше поставлених цілях, які компанія хоче досягти, впроваджуючи систему. Логічно пам'ятати про можливі ризики та витрати різних ресурсів. Перш за все, перед тим як

приймати рішення щодо запуску ІС, необхідно визначити те коло завдань, які повинна підтримувати інформаційна система. Для успішної роботи з інформаційною системою необхідно мати у розпорядженні техніку, яка здатна підтримувати роботу зі системою [45-46].

Технічне забезпечення включає в себе перш за все організаційну техніку на якій проводиться впровадження системи планування ресурсів ІТ-проєктів. Організаційна техніка призначена для реалізації технологій зберігання, подання та використання інформації, а також для виконання різних допоміжних операцій в рамках тих чи інших технологій інформаційної підтримки управлінської діяльності [47].

3.1.2 Архітектура програмного забезпечення

Ціль розробки програмного продукту реалізувати процес планування ресурсів ІТ-проєктів. На виході роботи програмного продукту користувач отримує побудований розклад (план) виконання завдань, що використовується у рамках діяльності компанії [49, 51].

Для відображення архітектури програмного забезпечення, нище наведено фрагмент діаграми класів на рисунку 3.2.

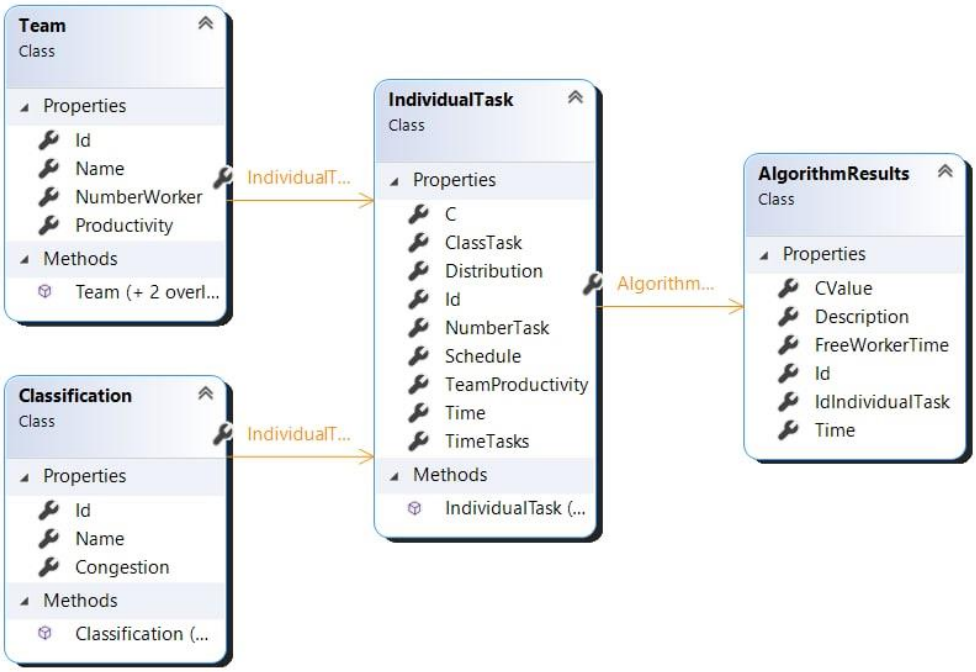

Рисунок 3.2 – Фрагмент діаграми класів програмного забезпечення

Згідно з результатами проведеного аналізу та враховуючи вищезазначені функції, було розроблено дерево цілей системи планування ресурсів IT-проєктів, що зображене на рисунку 3.3.

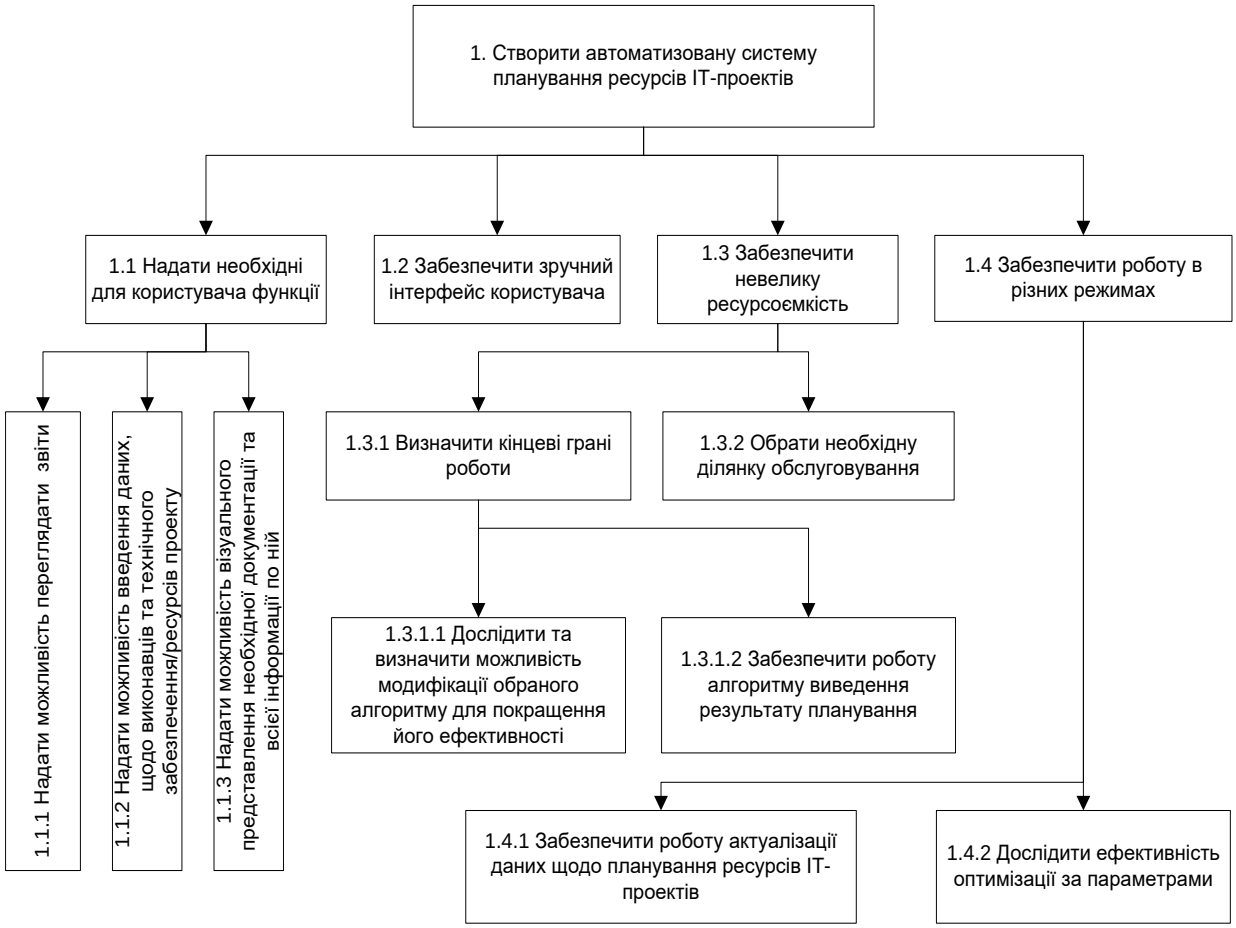

Рисунок 3.3 – Декомпозиція цілей системи

Інформаційне забезпечення системи – це набір елементів єдиної системи класифікації та уніфікованих систем ведення документації, схем інформаційних процесів, які повторюються в організації, і методології побудови даних. Нище наведено приклад, які блоки обробки проходить інформація, яка потрапила у систему, які блоки несуть відповідальність за ту чи іншу обробку, взаємодіють між собою. Схему обробки інформації представлено на рисунку 3.4.

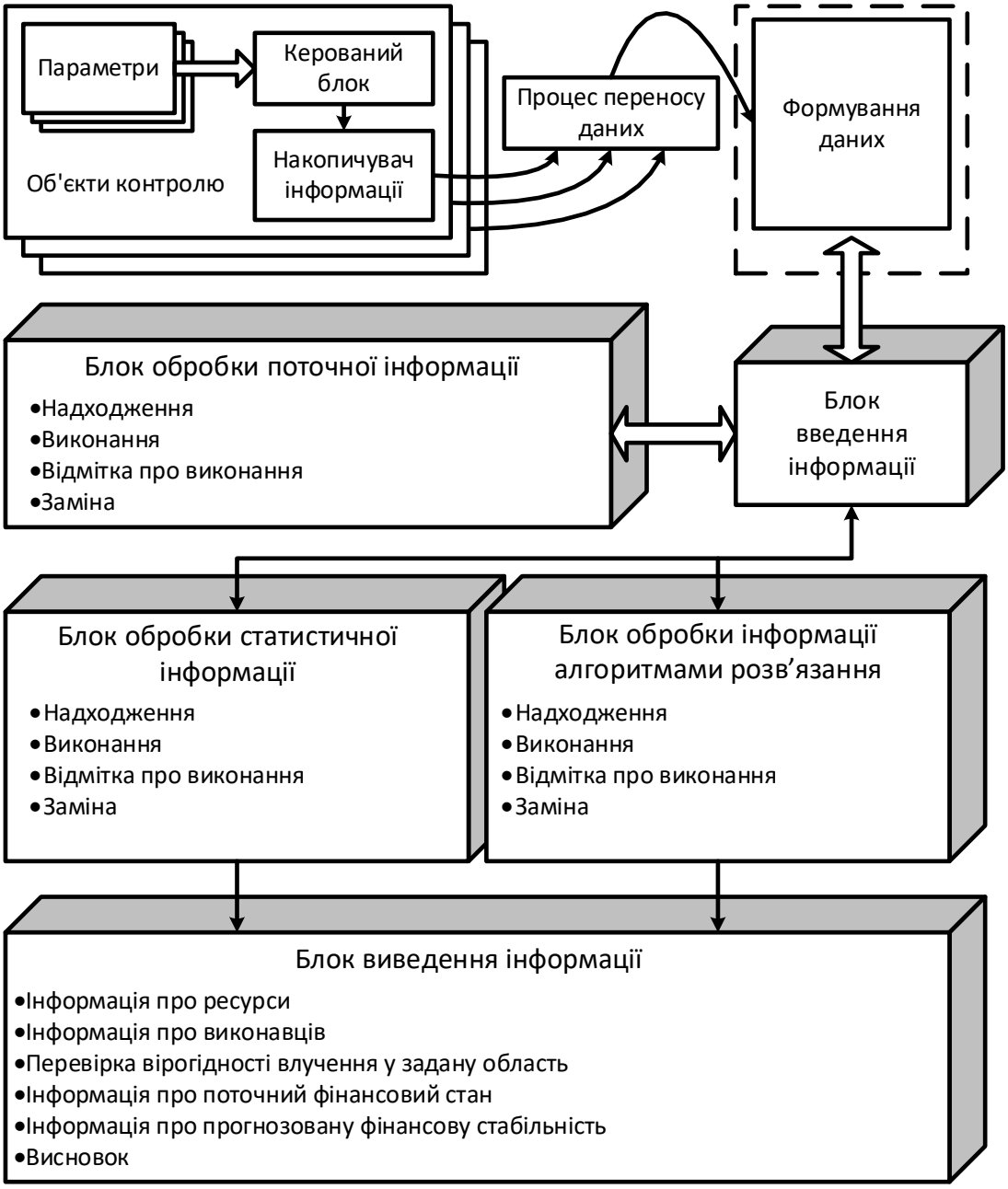

Рисунок 3.4 – Схема обробки інформації у розробленій системі

Задача обліку документації планування ресурсів IT-проєктів на етапі експлуатації шляхом розробки і впровадження інформаційної технології моніторингу документації за статистичними даними реалізацій визначальних параметрів в наш час є доволі актуальною [50].

Методи моніторингу і контролю є методами технічної діагностики і здатні не тільки забезпечити вирішення задач визначення поточного стану і

індивідуального планування ресурсів ІТ-проєктів, але й обробки та видачі інформації про стан об'єкту.

На рисунку 3.4 зображені головні функціональні модулі інформаційної системи, їхня взаємодія та зв’язки між ними. Кожен такий модуль відповідає за виконання певної функції в системі [53-54], але це не клас програмного забезпечення.

Функціональна структура програми представлена на рисунку 3.5.

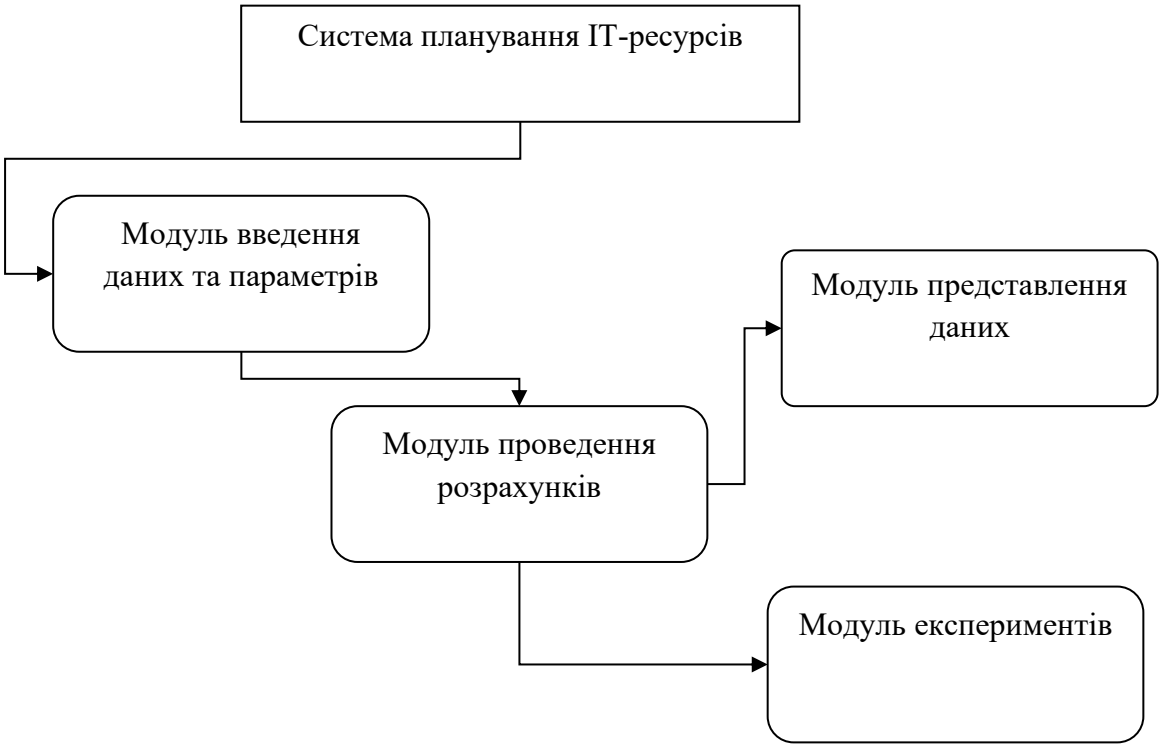

Рисунок 3.5 – Структурна схема програмного забезпечення

Модуль введення даних та параметрів

Цей модуль забезпечує введення даних по завданнях та виконавцях. Програма зберігає числові дані кожного показника для подальшого використання системою планування ресурсів ІТ-проєктів [55-57].

Модуль проведення розрахунків

Цей модуль відповідає за аналіз введених даних та проведення розрахунків розробленими алгоритмами вхідної задачі.

Модуль експериментів

Цей модуль відповідає за проведення експериментів у системі. Користувач має можливість проводити різноманітні експерименти для вхідних задач і використовувати отримані результати для досліджень. Нові показники будуть поступово доповнювати та змінювати остаточний результат.

Модуль представлення даних

Цей модуль відповідає за представлення отриманих результатів та виведення їх на екран користувача у текстовому вигляді. Чим більше буде введено вхідної інформації, тим детальніше будуть представлені результати [58-60].

Дана система має помітну відмінність від інших систем, за рахунок того, що розрахована на велике коло користувачів.

3.1.3 Користувацький інтерфейс

Користувацький інтерфейс програми має бути організованим зручним для користувача чином і відповідати стандартам дизайну інтерфейсів.

Інтерфейс складається з таких частин:

– головна сторінка;

– сторінка розв’язання однієї задачі та виведення результатів (розклад);

– сторінка з класифікатором задач для проведення експериментів. (експерименти);

– сторінка генерування задач (генератор задач).

Екранна форма програми для формування розкладу після введення даних для однієї задачі, що необхідно розв’язати, наведена на рисунку 3.6-3.7, де також різні екранні форми для роботи жадібного алгоритму та алгоритму локального пошуку.

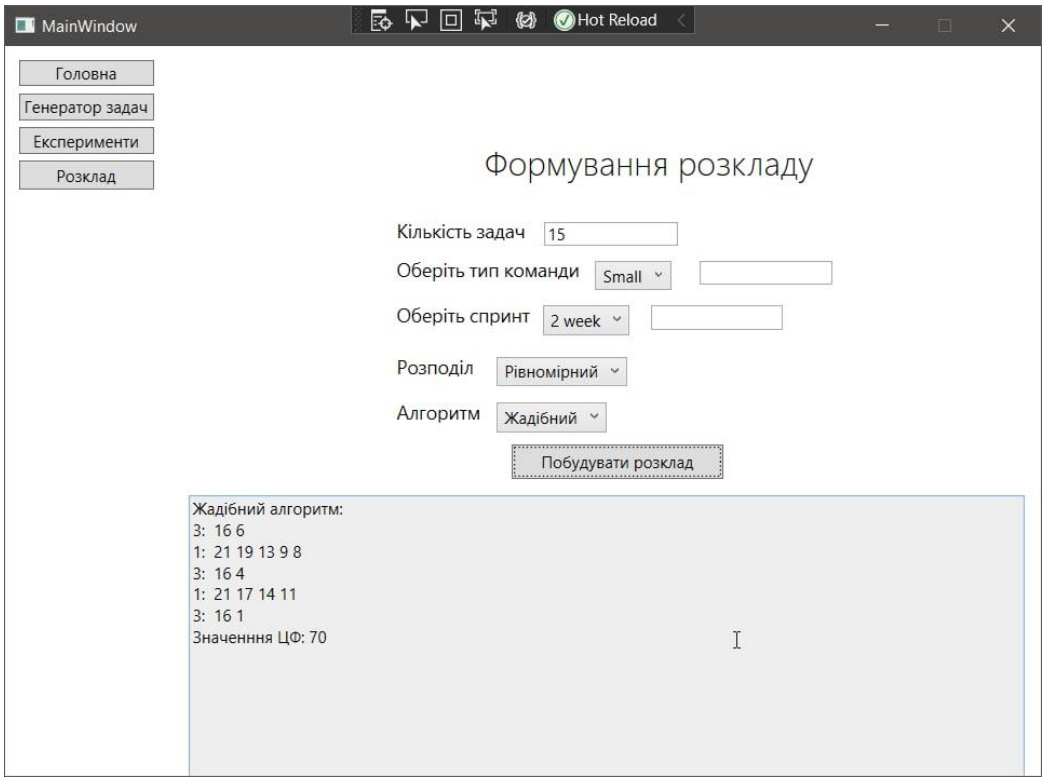

Рисунок 3.6 – Екранна форма для формаування розкладу задачі (жадібний алгоритм)

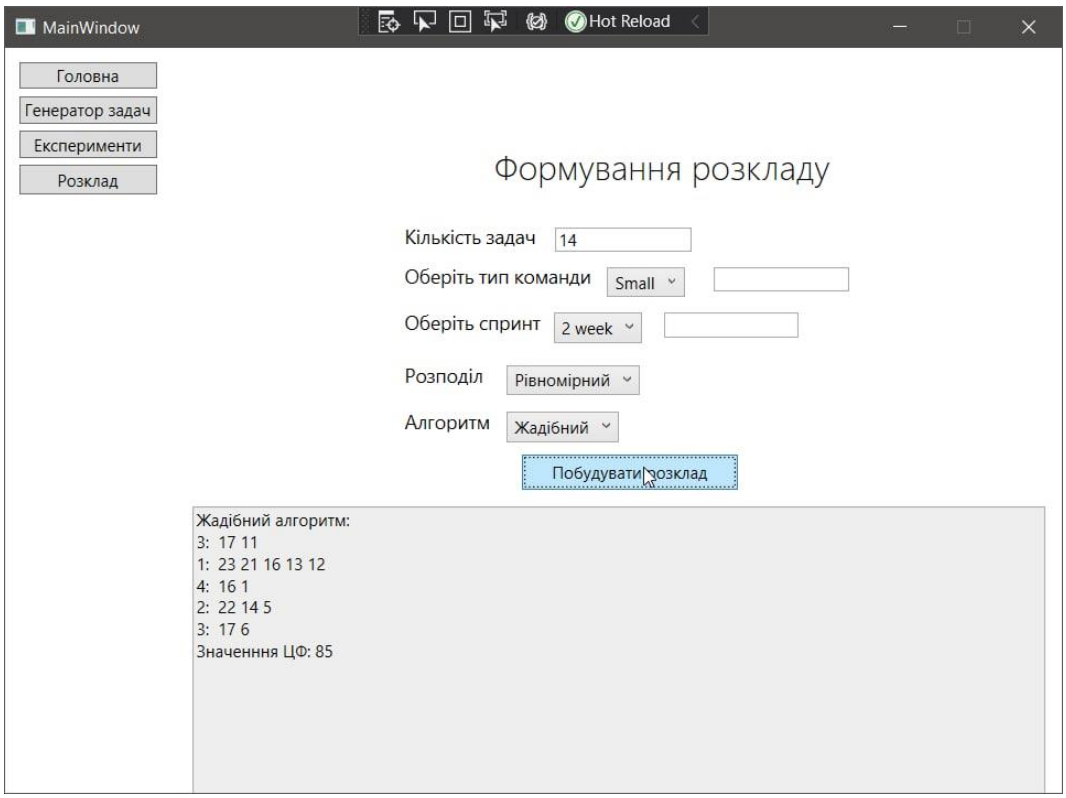

Рисунок 3.7 – Екранна форма для формування розкладу задачі (алгоритм локального пошуку)

Користувач має бути обізнаним в сфері ведення проєкту, так як він має всі повноваження у рамках системи, його дії будуть впливати на результат роботи системи [61-62].

Система виконує наступні вимоги та функції:

– надавати користувачам вводити показання (ресурси/виконавці), в формі введення даних;

– аналізувати введені показники і оброблювати їх, зберігаючи у вигляді, доступному для обробки програмою;

– проводити згортку значень, формуючи на основі множини оптимальні результати;

– виводити результати роботи в зручному для сприйняття і використання вигляді;

– розмежовувати доступ до введення різних типів інформації;

– не допускати введення значень, які можуть викликати збій в роботі системи;

– зберігати і завантажувати інформацію для подальшого використання;

– кожне вікно системи має містити в собі довідку про теоретичну інформацію, та інформацію щодо використання програми;

– програмний код системи має відповідати стандартам кодування;

– одночасно однією версією системи може користуватись безліч користувачів заходячи з різних ПК.

Наведемо нище екранні форми для генератора задач та експериментів на рисунках 3.8 та 3.9.

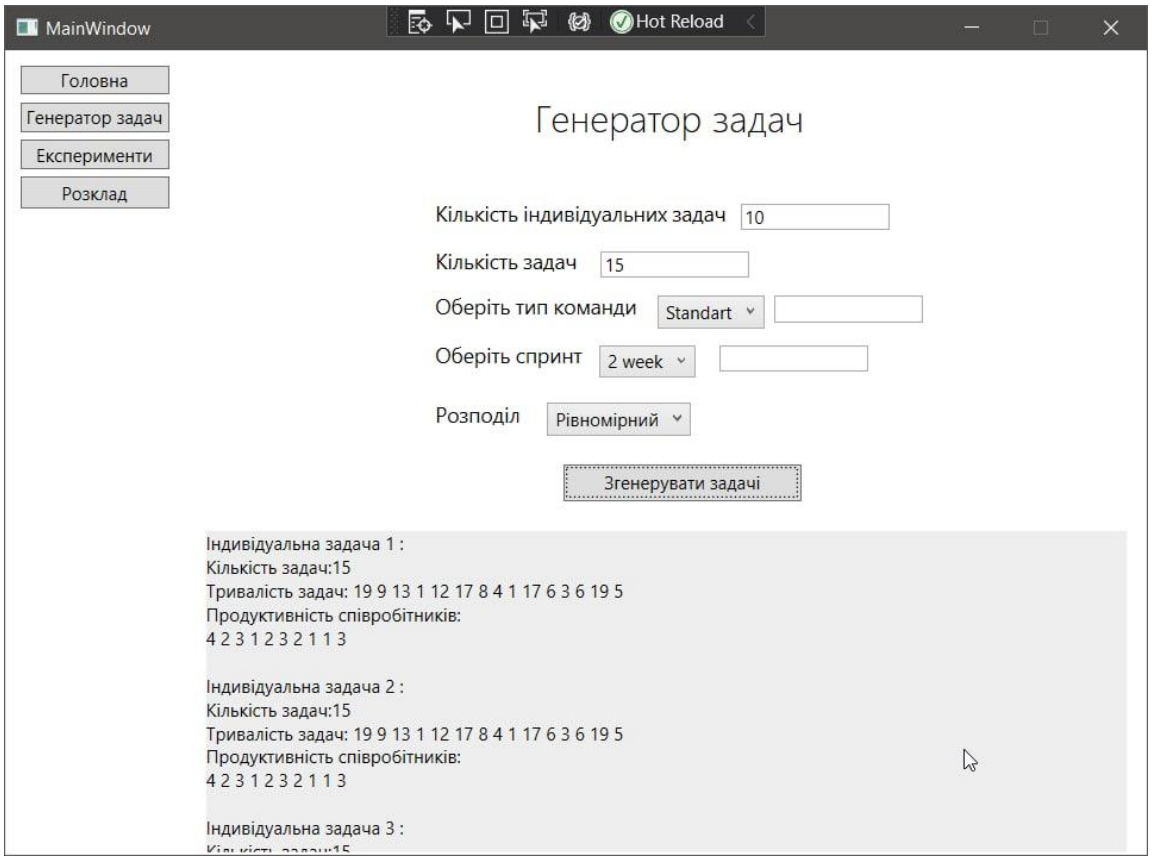

Рисунок 3.8 – Екранна форма генератора задач

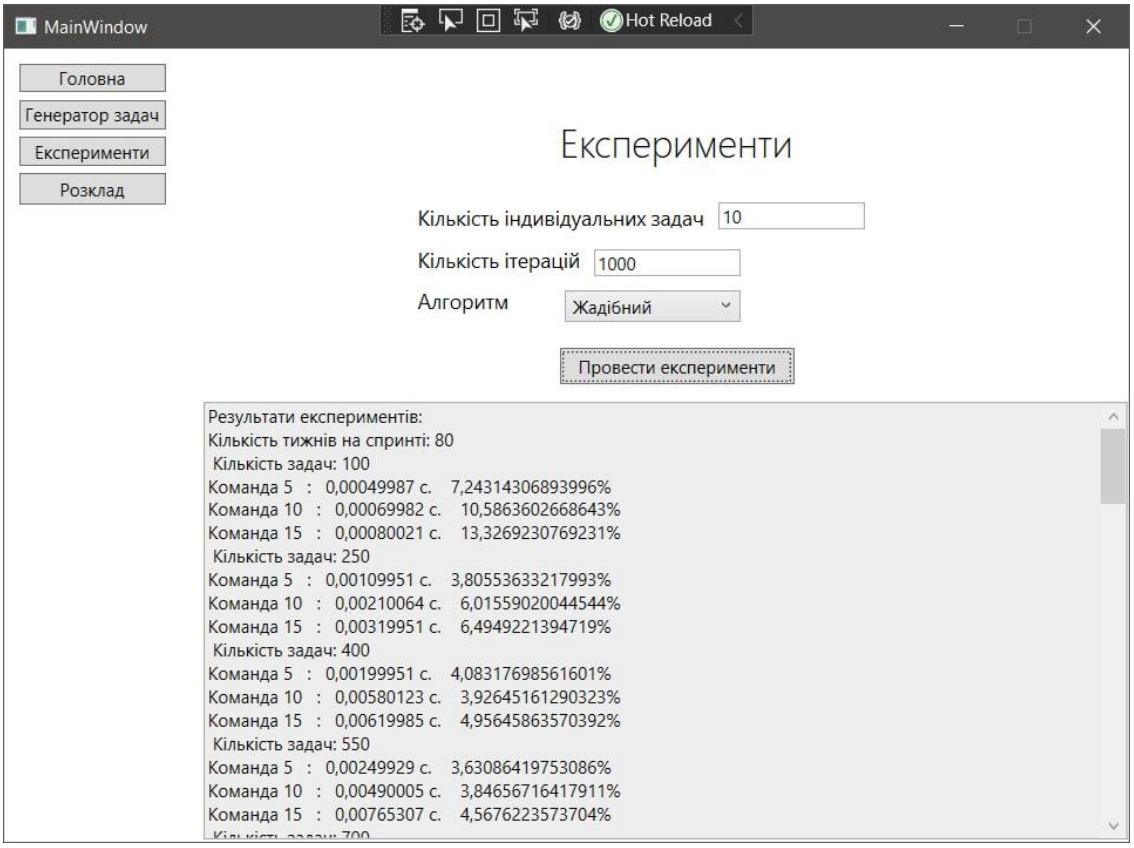

Рисунок 3.9 – Екранна форма проведення експериментів

Надійність. Програма має високий рівень надійності та готовність до роботи з помилками, що виконані користувачем[63].

Безпека даних. Програма має засоби для збереження безпеки даних: шифрування файлів [64].

Мобільність. Програма працює на ПК, з встановленою ОС Windows 7/10.

Зручність використання. Програма має мати зручний "дружній" інтерфейс, який дасть змогу користувачам використовувати систему без попереднього навчання роботи зі системою [65-66].

Усі користувачі мають доступ до роботи з базою даних, так як можуть вводити лише власну інформацію і не можуть нашкодити інформації інших користувачів.

3.1.4 Засоби розробки

Вибір того чи іншого програмного засобу визначається як специфікою розробки програмного забезпечення та його популярністю, так і фінансовими можливостями розробника.

У якості мови програмування було обрано C #.

Мова програмування C # є одинією з найбільш затребуваних мов в IT-галузі, використовується для створення як настільних (десктопних) програм, так і для веб-додатків. C # була створена спеціально для роботи з фреймворком (платформою) Microsoft .NET, що надає безліч можливостей і що спрощує створення додатків. Одним з основних компонентів пакету Microsoft .NET Framework є Common Language Runtime (CLR) загальномовне виконуюче середовище, яке компілює код програми на мові MSIL (в який компілюється код на мові C #) під час його виконання, а також надає MSIL-програмами (а отже , і програмами, написаним на мовах високого рівня, що підтримують .NET Framework) доступ до бібліотеки класів .NET Framework [67-68].

За допомогою C # і .NET Framework можна створювати класичні

настільні Windows Forms і Windows Presentation Foundation додатки, веб-додатки (використовуючи технологію ASP.NET), компоненти для розподілених додатків і доступу до баз даних. C # надає засоби для розробки [69] практично будь-якого типу програмного забезпечення для платформи Windows.

У C # розроблявся дружній інтерфейс, для роботи з системою тестування. Також створення робочої області типу панелі управління «кнопок» [70].

Для того, щоб програма виконувала наведені функції, наприклад, обчислювала, виводила результат, реагувала на дії користувача, наприклад, на натискання кнопок, вибір рядків зі списку, необхідний програмний код [71].

В якості інструментального середовища проєктування використовується Rational Software Architect [72].

3.2 Опис отриманих результатів розв’язання та проведених експериментів

Проведено експериментальні дослідження для розв’язання поставленої задачі. Нище наведено детальний опис отриманих результатів.

Оскільки у магістерській дисертації було представлено багатокретиріальну задачу теорії розкладів, то перш за все було визначено певний класифікатор задач, у основі якого виділено розподілення вхідних даних завдяки генератору різних значень.

Під час проведення експериментів були виділені основні параметри, які впливають на результат та час роботи алгоритмів.

Параметр 1 - розмір спринта. Так як реалізація завдань проходить з використанням підходу Scrum, то планування виконання завдань відбувається за допомогою спринтів. У таблиці 3.2 наведено позначення та розмір спринта відповідно.

Таблиця 3.2 – Класифікація спринтів за розміром

Позначення у класифікаторі	**Тривалість спринта у тижнях**	**Потужність спринта на 1 людину**
2 weeks	2 тижні	80 годин
3 weeks	3 тижні	120 годин
4 weeks	4 тижні	160 годин

Відповідно, у залежності від того, скільки учасників команди подається на вхід – відповідно від того рахується потужність спринта на всю команду.

Параметр 2 – розмір команди. Оскільки працюємо зі Scrum, зазвичай ці команди невеликих розмірів, проте є певні діапазони для кількості учасників команди. Діапазони стосовно кількості учасників команди наведено у таблиці 3.3.

Таблиця 3.3 – Класифікація команд за розміром

Позначення у класифікаторі	**Кількість учасників**
Small	1-5 учасників
Standart	6-10 учасників
Big	11-15 учасників

Параметр 3 – кількість завдань на спринт. Це значення було використане у діапазоні від 100 до 1000 завдань, так як завдання можуть бути за тривалістю виконання дуже великими, але їх може бути доволі мало на вході у систему. І навпаки. Нище у таблиці 3.4 наведено перелік значень кількості завдань, які використовувались для проведення експериментів.

Таблиця 3.4 – Перелік значень кількості завдань для експериментів

Кількість завдань
100
150
400
550
700
800
1000

Перейдемо до експериментів.

Експеримент 1.

Для проведення експериментів командам надавалися завдання кількістю: 100, 150, 400, 550, 700, 850 та 1000. Для жадібного та алгоритму локального пошуку визначався відсоток відхилення значення отриманого розкладу від очікуваного значення і цільової функції.

На вході маємо:

– спринт – 2 тижні (weeks);

– кількість учасників команди – змінна;

– кількість задач – змінна.

На виході отримуємо:

Значення відхилення у відсотках (%) отриманого розкладу від очікуваного значення і цільової функції.

Результати проведення цього експерименту наведено на рисунках 3.10 та 3.11, а отримані дані наведено у таблиці 3.5.

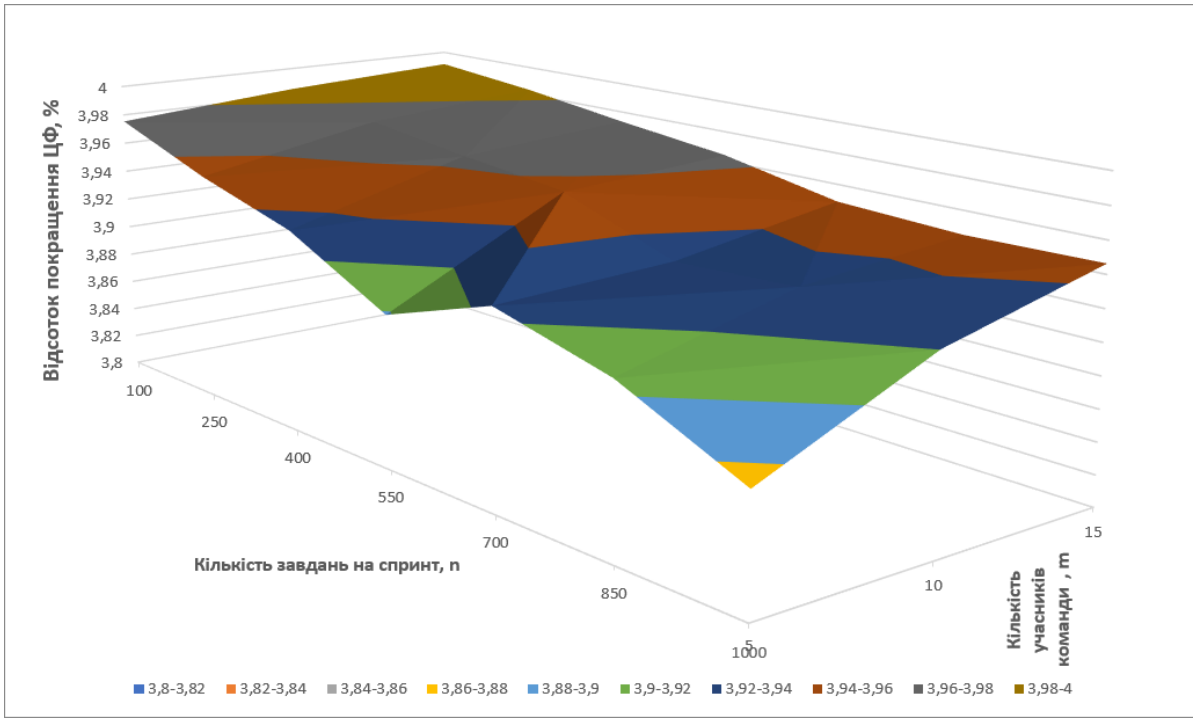

Рисунок 3.10 – Графік результатів роботи алгоритму локального пошуку з відображення відсотку покращення значення цільової функції

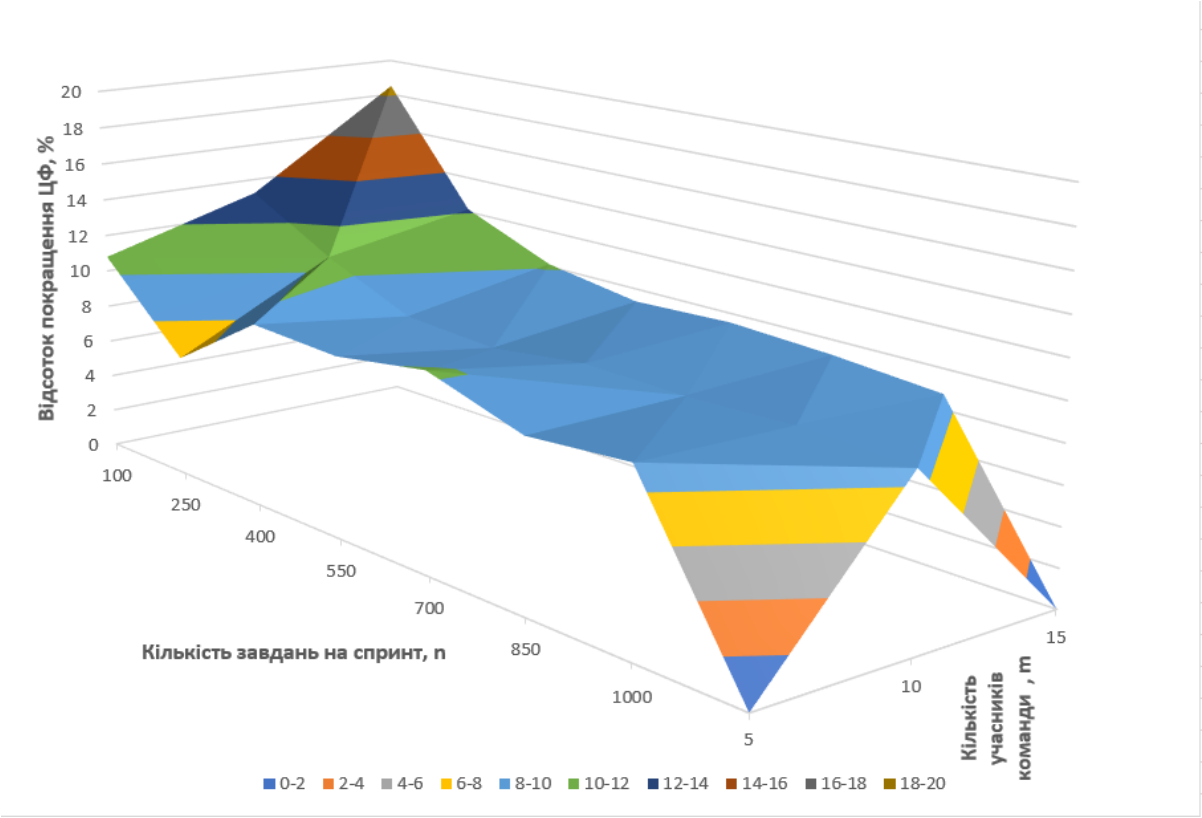

Рисунок 3.11 – Графік результатів роботи жадібного алгоритму з відображення відсотку покращення значення цільової функції

Таблиця 3.5 - Відхилення у % значень ЦФ від очікуваних результатів для жадібного алгоритму

	Кількість учасників команди		
Клас спринта	5	10	15
2 weeks	10,7924	13,21922	18,48312
	6,456402	10,66193	12,14874
	9,683307	8,630869	10,13338
	9,45012	8,318408	9,310358
	10,22611	8,96475	9,515012
	8,67269	8,839579	9,227035
	9,12331	9,026354	8,759698
3 weeks	10,33026	8,703937	18,50117
	8,688955	11,33903	11,98379
	9,313208	9,281798	10,15748
	9,690909	9,111662	9,774733
	9,955739	8,622708	8,810604
	8,818788	8,69741	9,272911
	8,600035	9,312917	8,837391
4 weeks	10,37504	13,46775	17,64857
	10,03943	10,25907	12,47046
	9,189422	8,414399	9,807089
	9,139294	9,551259	9,019445
	8,05643	8,743343	9,323892
	9,269489	8,740028	8,880613
	8,810404	9,350912	8,428605

Жадібний алгоритм мав найбільший відсоток відхилення отриманого значення цільової функції, проте отримані значні переваги у часі роботи

алгоритму. Алгоритм надає такі значенні цільової функції, що відхилення значення цільової функції від отриманих результатів не перевищує 3-4%.

Експеримент 2

Для проведення експериментів командам надавалися завдання кількістю: 100, 150, 400, 550, 700, 850 та 1000.

На вході маємо:

– спринт – 4 week;

– кількість учасників команди – змінна;

– кількість задач – змінна.

На виході отримуємо:

Значення часу роботи алгоритму у ms над цим класом задач.

Результати експерименту представлено на рисунках 3.12 та 3.13, а отримані значення наведено у таблиці 3.6.

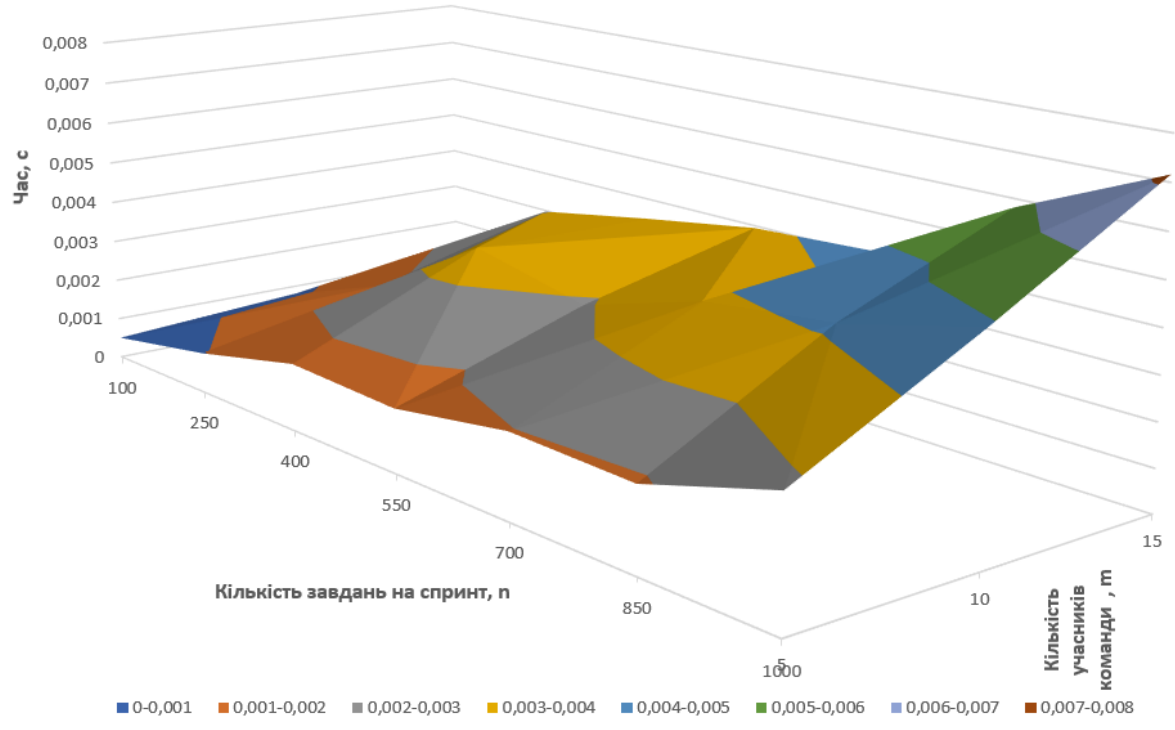

Рисунок 3.12 – Графік результатів часу роботи жадібного алгоритму з урахуванням параметрів

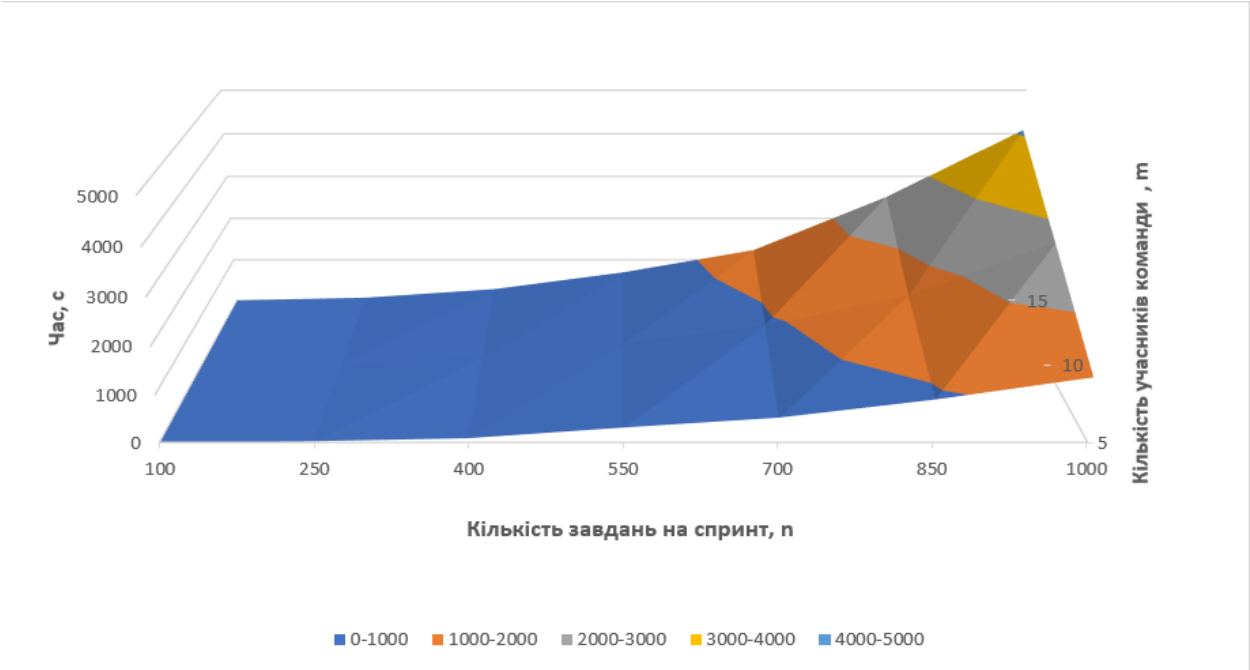

Рисунок 3.13 – Графік результатів часу роботи алгоритму локального пошуку з урахуванням параметрів

Таблиця 3.6 – Час роботи жадібного алгоритму (у секундах)

	Кількість учасників команди		
Клас спринта/ кількість завдань	5	10	15
2 weeks			
100	0,000498	0,000802	0,00112
250	0,000967	0,001805	0,00297
400	0,001589	0,003515	0,003478
550	0,001483	0,002909	0,003942
700	0,001955	0,003813	0,004103
850	0,001887	0,004164	0,005809
1000	0,00282	0,004922	0,007163

Продовження таблиці 3.6.

	Кількість учасників команди		
Клас спринта/ кількість завдань	5	10	15
3 weeks			
100	0,000248	0,000418	0,000544
250	0,00053	0,000941	0,001483
400	0,001005	0,002123	0,00305
550	0,001438	0,00256	0,004227
700	0,001767	0,003274	0,005468
850	0,002136	0,003289	0,005048
1000	0,002515	0,005245	0,006621
4 weeks			
100	0,000337	0,000529	0,00069
250	0,000683	0,001269	0,001723
400	0,001077	0,002042	0,003048
550	0,001206	0,002246	0,003228
700	0,001721	0,003332	0,005284
850	0,002067	0,004274	0,005971
1000	0,002182	0,003947	0,005576

Щодо швидкості роботи: вище представлено результати досліджень, що відповідають швидкості роботи алгоритмів.

Експерименти показують, що алгоритм локального пошуку добре працює у знаходженні кращих рішень, незважаючи на те, що час виконання ним ітерацій для пошуку кращого рішення більший. Це зумовлено також ще тим, що в основі алгоритму локального пошуку лежить саме жадібний алгоритм, і на кожній ітерації він повторюється за конкретних умов.

Експеримент 3

Результати збіжності двох алгоритмів показують, що алгоритм локального покушу плавно сходиться після 210 розмірності, тоді як жадібний сходиться після 250 та 270 відповідно. Тобто алгоритм локального пошуку, з точки зору пошуку кращого рішення, здатний знайти це рішення за меншу кількість ітерацій, аніж жадібний алгоритм.

Висновки до розділу

У третьому розділу здійснено опис програмного та технічного забезпечення системи планування IT-ресурсів. Розкрито структуру програмного додатку, здійснено опис процесу взаємодії з програмним забезпеченням.

Було наведено ряд експериментів та їх результатів, які були проведені у ході дослідження. Розглянуто евристичні алгоритми розв’язання: жадібний алгоритм та алгоритм локального пошуку. Для задач великої розмірності, час роботи алгоритмів також суттєво відрізняється: жадібний алгоритм працює швидше у 9000 разів аніж алгоритм локального пошуку. Це зумовлено тим, що для 100 завдань, жадібний алгоритм будує розклад для кожного виконавця і на цьому його робота завершується, а алгоритм локального пошуку має заданий ліміт ітерацій для повторення перестановок, що зумовлює великі кількість операцій для створення розкладу для 100 завдань. Але при цьому не гарантовано отримання кращого результату за допомогою жадібного алгоритму, тоді як результати роботи алгоритму локального пошуку надають результати з відхиленням на 3-4% від значення цільової функції. Представлено тривимірні графіки, які було побудовані на основі отриманих результатів досліджень.

Дана програма може буде застосована менеджерами та індивідуальними користувачами для автоматизованого планування ресурсів IT-проєктів, що

скоротить час розрахунків та дозволить спостерігати за рухом по кожному окремому параметру.

4 РОЗРОБКА СТАРТАП-ПРОЄКТУ

4.1 Опис ідеї проєкту

Опис ідеї старпат-проєкту, напрямки застосування та вигоди для користувачів наведено у таблиці 4.1.

Таблиця 4.1 – Опис ідеї стартап-проєкту

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**
Розробка інформаційної системи планування ІТ-ресурсів	1. Підвищення ефективності планування ІТ-ресурсів	Нові механізми застосування системи планування ІТ-ресурсів
	2. Оптимізація швидкості прийняття рішень щодо реалізації проєкту	Продуктивність застосування системи планування ІТ-ресурсів
	3. Здійснення обґрунтованого розподілу ресурсного забезпечення проєкту	Зниження вартості застосування системи планування ІТ-ресурсів

Конкурентами є аналогічні методи та механізми планування ІТ-ресурсів. Інформаційна система планування ресурсів ІТ-проєктів дозволяє підвищити оперативність в роботі; забезпечити простоту і зручність розрахунків; зменшити кількість помилок при планування ресурсів ІТ-проєктів; підвищити ступінь достовірності інформації; зменшити трудові витрати на обробку інформації.

Довгостроковими перспективами є:

– збільшення кількості клієнтів, що будуть використовувати інформаційну систему планування ресурсів ІТ-проєктів.

– додавання новітніх механізмів реалізації дієвого планування ресурсів ІТ-проєктів.

Потреби в стартовому фінансуванні:

Стартовий капітал = 10000 грн

У таблиці 4.2 наведено техніко-економічні характеристики ідеї.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проєкту розробки інформаційної системи планування ресурсів ІТ-проєктів

№ з/п	**Техніко-економічні характеристики ідеї**	**(потенційні) товари/концепції конкурентів**				**W (слабка сторона)**	**N (нейтральна сторона)**	**S (сильна сторона)**
		Мій проєкт	Конкурент 1	Конкурент 2	Конкурент 3			
1.	Бюджетне фінансування	виділення власних коштів	виділення бюджетних коштів на розробку	розробка комерційна	виділення власних коштів	повна відсутність фінансування	часткова бюджетне фінансування	виділення бюджетних коштів на розробку
2.	Використання сучасної техніки	використано найсучаснішу техніку для проєктування	використано техніку, яка є застарілою	використано техніку, яка є застарілою	використано найсучаснішу техніку для проєктування	використано найсучаснішу техніку для проєктування	часткова комплектація технікою	використано техніку, яка є застарілою

Продовження таблиці 4.2.

№ з/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
3.	Належна матеріально-технічна база	розробка системи здійснюється на власній техніці з використанням власних коштів	розробка здійснюється на базі бюджетної установи	розробка здійснюється на базі інформаційного центру	розробка здійснюється на базі інформаційного центру	розробка здійснюється на базі інформаційного центру	розробка здійснюється на базі бюджетної установи	розробка системи здійснюється на власній техніці з використанням власних коштів
4.	Налагоджена система реклами продукту	реклама відсутня	реклама присутня	реклама відсутня	реклама присутня	не має реклами	частково реклама	реклама присутня

Продовження таблиці 4.2.

№ з/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
5.	Високий рівень розробки	запропоновані методи та алгоритми є досконалими	на базі розробки є помилки вона потребує доробок	запропоновані методи та алгоритми є досконалими	на базі розробки є помилки вона потребує доробок	на базі розробки є помилки вона потребує доробок	розробко є майже досконалою	запропоновані методи та алгоритми є досконалими
6.	Професіонали програмісти	Головний розробник - студент	Головний розробник - група професіоналів	Головний розробник – програм професіонал	Головний розробник – програм професіонал	Головний розробник - студент	Головний розробник – програм професіонал	Головний розробник - група професіоналів

4.2 Технологічний аудит ідеї проєкту

Визначення технологічної здійсненності ідеї проєкту передбачає аналіз таких складових у таблиці 4.3:

– за якою технологією буде виготовлено товар згідно ідеї проєкту?

– чи існують такі технології, чи їх потрібно розробити/доробити?

– чи доступні такі технології авторам проєкту?

Таблиця 4.3 – Технологічна здійсненність ідеї проєкту розробки інформаційної системи планування ресурсів ІТ-проєктів

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Дослідження особливостей формування інформаційної системи планування ресурсів ІТ-проєктів	Технологія 1 (технологія надання послуги)	потрібно розробити	доступні
2		Технологія 2 (наявність бази досліджень)	наявні	доступні
3		Технологія 3 (база проведення досліджень (випробувань))	потрібно розробити	доступні
4		Технологія 4 (оформлення результатів дослідження)	потрібно розробити	доступні
Обрана технологія реалізації ідеї проєкту розробки інформаційної системи планування ресурсів ІТ-проєктів: є можливою				

4.3 Аналіз ринкових можливостей запуску стартап проєкту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проєкту, та ринкових загроз, які можуть перешкодити реалізації проєкту, дозволяє спланувати напрями розвитку

проєкту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проєктів-конкурентів. Цю інфоормацію представлено у таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	**Показники стану ринку (найменування)**	**Характеристика**
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	55000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	не має
5	Специфічні вимоги до стандартизації та сертифікації	ДСТУ 2844-94 ДСТУ ISO 9000-2007 (ISO 9000:2005, IDT) ДСТУ ITU-T G.957:2010
6	Середня норма рентабельності в галузі (або по ринку), %	32

На основі проведеного дослідження є можливість стверджувати про привабливість проєкту для входження на ринок за попереднім оцінюванням.

У таблиці 4.5 наведено перелік характеристик потенційних клієнтів даного стартап-проєкту, а у таблиці 4.6 наведено відповідно фактори загроз та можливостей відповідно. У таблиці 4.7 наведено ступеневий аналіз конкуренції на ринку.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	**Потреба, що формує ринок**	**Цільова аудиторія (цільові сегменти ринку)**	**Відмінності у поведінці різних потенційних груп клієнтів**	**Вимоги споживачів до товару**
1	Дослідження особливостей формування інформаційної системи планування ресурсів ІТ-проєктів	ІТ-центри, телекомунікаційні центри	планування ресурсів ІТ-проєктів, підвищення ефективності планування, скорочення часу на планування, підвищення ефективності роботи підприємства	відповідність ДСТУ 2844-9 Обов’язкова наявність сертифікатів

Таблиця 4.6 – Фактори загроз

№ п/п	**Фактор**	**Зміст загрози**	**Можлива реакція компанії**
1	Агресивність конкурентів на ринку	Вплив конкурентів на систему	може порушити налагоджену систему розповсюдження
2	Нестабільність політичної ситуації як на території України так і в світі	Варіювання курсу валют	може порушити надійну систему постачальників

Продовження таблиці 4.6.

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
3	Висока вартість інформаційної системи	Збільшення ціни на систему	підвищить агресивність конкурентів
4	Економічні складності при розробці	Повна відсутність фінансування або його мінімальне фінансування	може порушити фінансове забезпечення компанії

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Тривале існування інформаційної системи планування ресурсів ІТ-проєктів	тривале існування на ринку інформаційної системи планування ресурсів ІТ-проєктів	на ринку дає можливість виходу на нові ринки
2	Моніторинг потреб споживачів інформаційної системи планування ресурсів ІТ-проєктів	розуміючи потреби споживачів, розширювати діапазон пр інформаційної системи планування ресурсів ІТ-проєктів одукції, що випускається.	розширення діапазону інформаційної системи планування ресурсів ІТ-проєктів

Продовження таблиці 4.7.

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3	Лібералізація торговельних бар'єрів при розповсюдженні інформаційної системи планування ресурсів ІТ-проєктів	робота менеджменту	призведе до поліпшення налагодженої системи розповсюдження інформаційної системи планування ресурсів ІТ-проєктів
4	Висока вартість інформаційної системи планування ресурсів ІТ-проєктів в порівнянні з ключовими конкурентами	встановлення високої ціни на інформаційну систему планування ресурсів ІТ-проєктів	ускладнює вихід на нові ринки інформаційної системи планування ресурсів ІТ-проєктів
5	Стабілізація бізнес-середовища при реалізації інформаційної системи планування ресурсів ІТ-проєктів	формування стабільного середовища при реалізації інформаційної системи планування ресурсів ІТ-проєктів	за рахунок стабілізації бізнес-середовища можна поліпшити фінансове забезпечення компанії

У таблиці 4.8 наведено ступеневий аналіз конкуренції на ринку, а у таблиці 4.9 описано аналіз конкуренції в галузі за М. Портером.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1. Вказати тип конкуренції - монополія/олігополія/ монополістична /чиста	Локальний/національний бізнес. Глобальні сили є не досить вагомими по відношенню до локальних сил, які визначаються наявністю сертифікації, відповідності держ нормам і стандартам, регулюванням інформаційної галузі державою.	працює в рамках підвищення параметрів процесу планування ресурсів ІТ-проєктів
2. За рівнем конкурентної боротьби- локальний/національний/…	Локальний	Ведучи конкуренцію на локальному рівні, компанії необхідно прикласти належні зусилля для охоплення всього ринку щодо реалізації інформаційної системи планування ресурсів ІТ-проєктів

Продовження таблиці 4.8.

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	Внутрішньогалузева. Конкуренція на ринку ведеться в інформаційній галузі України	Необхідно зосередити зусилля на пошуку конкурентних переваг, які дозволять компанії займати стійкі конкурентні позиції
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	Товарно-родова. Конкуренція на рівні технології задоволення потреб. Існує конкуренція з іншими моделями, алгоритмами	методи параметрів інформаційної системи планування ресурсів ІТ-проєктів водночас веде конкурентну боротьбу як з товарами-субститутами

Продовження таблиці 4.8.

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
5. За характером конкурентних переваг - цінова / нецінова	Нецінова. При виборі алгоритмів та методів споживач звертає увагу на ефективність методів та рівень якості інформаційної системи планування ресурсів IT-проєктів.Цінова. Для значної частки споживачів ціна є визначальною при виборі інформаційної системи планування ресурсів IT-проєктів	Головною конкурентною перевагою є унікальність позиціонування інформаційної системи планування ресурсів IT-проєктів
6. За інтенсивністю - марочна/ не марочна	Марочна	Диференціація методів та моделей за мотивом задоволення необхідних потреб споживачів

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Процес**	**Висновки**
Прямі конкуренти в галузі	Навести перелік прямих конкурентів	На ринку спостерігається тенденція до скорочення кількості інформаційних систем направлених на планування ІТ-ресурсів і посилення конкуренції на ринку. Вступ України до СОТ відкрив дорогу іноземним виробникам інформаційних систем планування ресурсів ІТ-проєктів. Великі компанії з іноземним капіталом постійно збільшують контрольовану ними частку ринку, поглинаючи конкурентів.
Потенційні конкуренти	Визначити бар’єри входження в ринок	Бар'єри входу на ринок є порівняно незначними. Вартість організації бізнесу з виробництва сучасних інформаційних систем планування ресурсів ІТ-проєктів сягає 120 тис. дол. Обов’язковою є сертифікація продукції (інформаційної системи планування ресурсів ІТ-проєктів).
Постачальники	Визначити фактори сили постачальників	Існує чітка залежність від постачальників як якості інформаційної системи планування ресурсів ІТ-проєктів. Також ціна кінцевої продукції залежить від рівня сертифікації.

Продовження таблиці 4.9.

Складові аналізу	**Процес**	**Висновки**
Клієнти	Визначити фактори сили споживачів	Споживачі мають широку географію і проживають переважно у містах. Покупка програмних додатків та алгоритмів реалізації планування ресурсів ІТ-проєктів часто носить імпульсний характер.
Товари-замінники	Фактори загроз з боку замінників	Посилилася конкуренція зі сторони товарів субститутів – інших видів методів та алгоритмів планування ресурсів ІТ-проєктів, за рахунок збільшення асортименту останніх та появи нових для ринку категорій.

Отже, відповідно до наведеного вище аналізу головними силами, які діють на конкуренцію в галузі є постачальники та споживачі. Також в силу розвитку ринку все більшого значення набуває інтенсивність конкуренції між існуючими конкурентами та загроза зі сторони товарів-субститутів [75-80].

Таким чином в межах структурного підходу до аналізу конкуренції тип конкуренції – *монополістична конкуренція*.

У таблиці 4.10 наведено пояснення та опис факторів конкурентоспроможності.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування вибору
1	Частка ринку	Враховуючи той факт, що тип родового середовища в галузі – консолідований ринок, тобто існує група компаній, які контролюють разом понад 40% ринку, а також те, що інтенсивність суперництва між діючими конкурентами при низьких темпах зростання ринку є однією з головних сил, одним з найважливіших факторів конкурентоспроможності виступає частка ринку, яку займає виробник. В таких умовах чим більше частка ринку, тим більшими ринковими можливостями володіє виробник.
2	Ціна	Чим вигіднішою є ціна для споживача, тим вірогідніше його вибір.
3	Асортимент	В умовах збільшення інтенсивності між існуючими конкурентами завоювання споживачів відбувається за рахунок нових методів та алгоритмів.
4	Доступ до каналів розподілу	Споживач далеко не завжди проявляє прихильність до певної категорії розробників і дуже схильний до експериментів. В цьому випадку завоювати лояльність споживача дуже складно і ще складніше її утримати. Тому для компаній-виробників ключовими чинниками успіху стає сильна дистрибуція, якісний торговий маркетинг і налагоджена система логістики.
5	Торговий маркетинг	

Продовження таблиці 4.10.

№	Фактор конкурентоспроможності	Обґрунтування вибору
6	Рівень диференціації ТМ	В умовах ведення конкурентної боротьби на споживчому ринку, де попит є ірраціональним та існує велика кількість виробників і розробників при фактично відсутній різниці між товарами, що пропонуються, ключовим фактором успіху є здатність чітко диференціювати ТМ від ТМ конкурентів, надаючи споживачеві унікальну цінність.
7	Репутація виробника	Якщо компанія має бездоганну репутацію, особливо у сфері якості своєї продукції, то рівень довіри до неї зростає. Також репутація виробника важлива при виході на ринок з новими товарами, або при виході на нові сегменти, що полегшує позитивне сприйняття новинок.
8	Рівень лояльності до бренду	Чим вище рівень лояльності, тим більше компанія має прихильних, а значить постійних споживачів.
9	Унікальність позиціонування	В умовах монополістичної конкуренції, коли фактор диференціації ТМ є ключовим засобом ведення конкурентної боротьби, важливим є створення та підтримання унікального позиціонування, що створює певний захист від конкурентних зіткнень.

Кінець таблиці 4.10.

№	Фактор конкурентоспроможності	Обґрунтування вибору
10	Маркетинговий бюджет	Від розміру маркетингового бюджету залежить здатність здійснювати маркетингову стратегію підприємства. Маркетингові заходи мають забезпечувати інші конкурентні переваги такі, як рівень диференціації, лояльності, репутація виробника, дистрибуція та просування в торгових точках.

Висновки до розділу

У четвертому розділі здійснено розробку стартап проєкту створення інформаційної системи планування ресурсів. Здійснено аналіз конкурентного середовища, розкрито фактори впливу на систему, здійснено обґрунтування факторів конкурентоспроможності. На основі проведеного дослідження визначено, що головними силами, які діють на конкуренцію в галузі є постачальники та споживачі. Також в силу розвитку ринку все більшого значення набуває інтенсивність конкуренції між існуючими конкурентами та загроза зі сторони товарів-субститутуів.

Проєкт є цілісною системою для входження на ринок за попереднім оцінюванням.

ВИСНОВКИ

У магістерській дисертації здійснено розробку взаємопов'язаного комплексу моделей, методики і алгоритмів, як основи системи підтримки прийняття рішень, що надає менеджеру проєкту своєчасну інформацію про стан проєкту і забезпечує підтримку ухвалення рішень з управління ресурсами і календарним плануванням.

У роботі вирішені наступні завдання:

– досліджено та виявлено переваги та недоліки існуючих методик управління проєктами, програмних комплексів, що спеціалізуються по автоматизації процесів управління проєктами та їх підтримки, алгоритмів складання розкладу проєкту, використання засобів інтелектуалізації в управлінні ресурсами проєкту;

– розроблено методику управління ресурсами проєкту, що дозволяє виконувати розподіл завдань проєкту між ресурсами, складати розклад проєкту з урахуванням ризиків, виконувати моніторинг стану проєкту і приймати управлінські рішення про заміну ресурсів в проєкті.

Розроблена інформаційна система планування ресурсів IT-проєктів дозволяє підвищити оперативність в роботі; забезпечити простоту і зручність розрахунків; зменшити кількість помилок при планування ресурсів IT-проєктів; підвищити ступінь достовірності інформації; зменшити трудові витрати на обробку інформації.

Дана програма може буде застосована менеджерами та індивідуальними користувачами для автоматизованого планування ресурсів IT-проєктів.

На основі проведеного дослідження визначено, що головними силами, які діють на конкуренцію в галузі є постачальники та споживачі. Також в силу розвитку ринку все більшого значення набуває інтенсивність конкуренції між існуючими конкурентами та загроза зі сторони товарів-субститутуів.

У результаті роботи над магістерською дисертацією:

– на основі проведеного аналізу існуючих рішень було визначено, що існуючі системи надають можливість виконати розподіл ресурсів та побудувати модель реалізації цього проекту, враховуючи обмеження ресурсів, але неспроможні встановити загальні часові обмеження для цілого проекту під час планування з урахуванням індивідуальних показників кожного виконавця завдань;

– розроблено програмний продукт, який забезпечує побудову плану розподілу завдань між учасниками на основі розроблених ефективних алгоритмів розв’язання задачі планування ресурсів, які дозволяють будувати оптимальні або ж близькі до оптимальних плани. Досліджено ефективність розроблених методів. Для цього було проведено ряд експериментів, які базуються на результатах роботи жадібного алгоритму і алгоритму локального пошуку за різної розмірності задачі.

– було встановлено, що алгоритм локального пошуку здатний знаходити краще рішення за меншу кількість ітерацій, аніж жадібний алгоритм. Для задач великої розмірності, час роботи алгоритмів також суттєво відрізняється: жадібний алгоритм працює швидше у 9000 разів аніж алгоритм локального пошуку. Це зумовлено тим, що для 100 завдань, жадібний алгоритм будує розклад для кожного виконавця і на цьому його робота завершується, а алгоритм локального пошуку має заданий ліміт ітерацій для повторення перестановок, що зумовлює великі кількість операцій для створення розкладу для 100 завдань. Але при цьому не гарантовано отримання кращого результату за допомогою жадібного алгоритму, тоді як результати роботи алгоритму локального пошуку надають результати з відхиленням на 3-4% від значення цільової функції.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бізнес-планування : навч. посіб. / Т. Г. Васильців, Я. Д. Качмарик, В. І. Блонська, Р. Л. Лупак. –К. : Знання, 2013. – 173 с.

2. SGV.IN.UA. Management and Project Management: Theory and Practice [Електронний ресурс] – Режим доступу: https://sgv.in.ua/off-lifaq/25-suchasni-metodi-upravlinnya-proektami

3. Бродська А. О. Використання інформаційних технологій в управлінні проєктами підприємств [Електронний ресурс] / А. О. Бродська // Управління розвитком складних систем. – 2013. – Вип. 13. – С. 8-11. – Режим доступу: http://urss.knuba.edu.ua/files/zbirnyk-13/8-11.pdf

4. Вольфсон, Б. Гибкие методологии разработки / Б. Вольфсон [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: http://adm-lib.ru/books/10/Gibkie-metodologii.pdf (дата обращения: 09.09.2020).

5. Воробець С. Й. Створення автоматизованих інформаційних систем на засадах процесного підходу / С. Й. Воробець, В. П. Кічор, А. В. Симак // Менеджмент та підприємництво в Україні: етапи становлення і проблеми розвитку : збірник наукових праць ; відп. ред. О. Є. Кузьмін. – Львів : Вид-во Львівської політехніки, 2012. – С. 408–413.

6. Голоскокова А. О. Моделі та інформаційна технологія планування покращення якості процесу розробки програмного забезпечення: диссертація / Голоскокова А. О. - Національний технічний університет" Харківський політехнічний інститут", 2018.

7. Данчук В. Д. Специфіка впровадження agile методологій для проєктів розробки програмного забезпечення / Данчук В. Д., Луцюк Д. В. - Вісник Національного транспортного університету 24, 2011- с 346-350.

8. Джулій В.М. Методи та алгоритми розробки web-додатків /Джулій В.М. - Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка, 2017 — с. 107-14.

9. Кіфер В. М. СКРАМ-ефективний підхід при розробці ПЗ/ Кіфер В. М. - Зв'язок 4, 2012 — с. 10-12.

10. Коляда А. С. Эффективность использовани адаптивных подходов при разработке программного обеспечения/ Коляда А. С. - Інформаційні технології в освіті, науці та виробництві,2012 - с 29-33.

11. Кондрин А. В., Кукарцев А. В. Стратегия внедрения CALS-технологий // Вестник СибГАУ. 2011. № 3 (36). С. 210-214.

12. Марченко А. В. Методология управления проєктами Scrum как пример внедрения методологии AGILE / Марченко А. В. - Зв'язок 4, 2016 — с. 27-30.

13. Морозов В. В. Компоненти управління проєктами: навчальний посібник для самостійної роботи студентів магістратури по спеціальності 8.000003 "Управління проєктами" / В.В. Морозов. – К.: Університет економіки та права "КРОК", 2005. – 62 с.

14. Морозов В. В. Формування, управління та розвиток команди проєкту / В.В.Морозов, А.М.Чередніченко, Т.І. Шпильова. – Київ: Таксон, 2009. – 461с.

15. Морозов В. В.Інформаційні системи і технології в управлінні проєктами. Планування проєктів у MS Project: навчальний посібник / В.В.Морозов, О.Б.Данченко, О.І. Шаров. – К. : Університет економіки та права "КРОК", 2011. – 167 с.

16. Новаківський І. І. Проєктно орієнтована організаційна система управління як ціль еволюції проєктного менеджменту / І. І. Новаківський // Проблеми економіки та управління. – Львів : Вид-во Львівської політехніки, 2009. – No 640. – С. 163–174.

17. Ноздріна Л. В. Управління проєктами: Підручник / Л. В. Ноздріна, В. І. Ящук, О. І. Полотай; за заг. ред. Л. В. Ноздріної. – К.: Центр учбової літератури, 2010. – 432 с.

18. Открытые системы. Scrum: гибкое управление разработкой [Электронный ресурс]: Статья [б.м, б.г, б.и].– Режим доступа: – http://www.osp.ru/os/2007/04/4220063/ (дата обращения: 20.09.2020).

19. Рач В. А. Багаторівнева системна модель виявлення специфічних проявів якостей особистостей в діяльності з управління проєктами / В. А. Рач // Управління проєктами: стан та перспективи: матеріали IV міжнародної науково-технічної конференції. – Миколаїв: НУК, 2008. – С. 131–134.

20. Самсонов А. К. Управління проєктом при створенні мобільного додатку / Самсонов А. К. - InProject, Program, Portfolio Management, 2017 — с. 79-81 ІКС ОНПУ.

21. Сванідзе Л. Г. Проєктний підхід в управлінні підприємницькою діяльністю : автореф. дис. на здобуття наук. ступеня канд. екон. наук : спец. 08.06.02 “Підприємництво, менеджмент та маркетинг” / Л.Г.Сванідзе. – Київ, 2001. –17 с.

22. Семенов С. Г. Усовершенствованный способ масштабирования гибкой методологии разработки программного обеспечения / Семенов С. Г. - ВНТУ, 2017.

23. Семеріков С. О. Мобільне програмне забезпечення / Семеріков С. О. - Програмне забезпечення, 2010 — 156 с.

24. Столярик П. О. Особливості візуалізації розробки проєктів за Scrum методологією для розподілених команд : диссертація/ Столярик П. О. - ВНТУ, 2018.

25. Тернер Р.Дж. Области приложения проєктно- ориентированного управления / Р. Дж. Тернер // Управление проєктами и программами, – 2017. – №3(11). – С. 220 – 236.

26. Чабанюк Я. М. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проєкту / Чабанюк Я. М. - Інженерія програмного забезпечення 1.1, 2010 - № 24, с. 120-235

27. Шведа Н. М. Система управління проєктами в Україні / Н.М.Шведа, Н.Є.Юрик // Збірник тез доповідей IVМіжнародної науково-технічної конференції молодих учених та студентів «Актуальні задачі сучасних технологій», 25-26 л истопада 2015 року. – Т. : ТНТУ, 2015. – Том 2. – С. 246-247.

28. Шишкіна М. П. Якість програмних забезпечення для мобільних пристроїв / Шишкіна М. П. - Науковий часопис Національного педагогічного університету імені МП Драгоманова, 2017 — 150 с.

29. Шпак Н. О. Переваги використання інформаційно-комунікаційних технологій в Україні [Електронний ресурс] / Н. О. Шпак, О. І. Венгер // Вісник Національного університету "Львівська політехніка". – 2012. – No 727. – С. 461–467. – Режим доступу: http://ena.lp.edu.ua:8080/ bitstream/ntb/13914/1/67_461-467_Vis_727_Menegment.pdf

30. Каверина С.Ю., Башинская И.А. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ ПРОЄКТАМИ / Випуск # 10 / 2017

31. Кучер О. Д, Використання ІТ в управлінні проєктами / О. Д, Кучер // 2019. – Режим доступу. – http://dspace.mnau.edu.ua/jspui/bitstream/123456789/6706/1/studentresearchjournal162-29.pdf.

32. Бойко Н. І. Багатовимірне подання даних для управління ІТ-проєктами. Національний університет “Львівська політехніка”, 2015. URL: http://ena.lp.edu.ua/bitstream/ntb/29918/1/37_387-394.pdf (дата звернення: 20.09.2020)

33. Івченко І. Ю., Будорацька Т.Д. Розробка моделі розподілу ІТ-проєктів на підприємствах галузі інформаційних технологій //Маркетинг і цифрові технології – ТЕС: Одеса Том 2, №3, 2018, С. 64-76 http://mdtopu.com.ua/files/download/mdt2.3.2018-16.09_1.pdf

34. Філатова Т., Чернишов О. Методологія представлення соціальних проєктів ІТ-індустрії// Матеріали ІІІ Міжнародної конференції

«Комп’ютерна алгебра та інформаційні технології», Одеса, 20–25 серпня 2018 р./Одеський національний університет імені І.І. Мечникова, – Одеса 2018. – 198с (80-87с) http://confit.onu.edu.ua/content/CAIT-Odessa-2018.pdf

35. Бродська А. О. Використання інформаційних технологій в управлінні проєктами підприємств / А. О. Бродська // Управління розвитком складних систем. - 2013. - Вип. 13. - С. 8-11. - Режим доступу: http://nbuv.gov.ua/UJRN/Urss_2013_13_4.

36. Project Management with Dynamic Scheduling - Baseline Scheduling, Risk Analysis and Project Control [Електронний ресурс] – Режим доступу: https://link.springer.com/book/10.1007%2F978-3-642-40438-2

37. A competitive genetic algorithm for resource-constrained project scheduling. Sönke Hartmann. First published: 07 December 1998 [Електронний ресурс] – Режим доступу: https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1520-6750(199810)45:7%3C733::AID-NAV5%3E3.0.CO;2-C

38. An iterative scheduling technique for resource-constrained project scheduling. K.Y.LiR.J.Willis. 2010 [Електронний ресурс] – Режим доступу: https://www.sciencedirect.com/science/article/abs/pii/0377221792903209

39. Abrahamsson, P., Warsta, J., Siponen, M., Ronkainen, J.; New directions on Agile methods: A comparative analysis. In proc. Of the Intl. Conf. on Software Engineering. 2003.

40. Метод підйому (локального пошуку) [Електронний ресурс] – Режим доступу до ресурсу: https://studfile.net/preview/3766334/page:6/

41. Жадібні алгоритми [Електронний ресурс] – Режим доступу до ресурсу: https://dl.sumdu.edu.ua/textbooks/95351/522264/index.html

42. Agilemanifesto [Электронный ресурс]. URL: http://agilemanifesto.org/iso/ru/manifesto.html (дата обращения: 09.01.2020).

43. Andrei Cristian Spataru. “Agile Development Methods for Mobile Applications”. Master of Science Thesis submitted to Computer Science School of Informatics, University of Edinburgh. 2010.

44. Atlassian JIRA Software[Електронний ресурс] // Офіційний сайт. – Режим доступу: https://atlassian.com/software/jira

45. Ayob, N.Z., Hussin, R.C., Dahlan H.M.; Three layers design guideline for mobile application. International Conference on Information Management and Engineering, pp. 427-431. 2009.

46. Ben Morris. Book on a Software Engineering Model. “A Software Engineering Model for Mobile App Development”, http://static.bada.com/contents/blog/20110616/20-introduction_to_bada_app03.pdf.

47. Chen G. and Kotz D., “A Survey of Context-Aware Mobile Computing Research,” Hanover, NH, USA, Tech. Rep., 2000.

48. Cohn M. User Stories Applied / Cohn M. - Kindle Edition, 2017 — 303 p.

49. Colomo-Palacios R., Calvo-Manzano J., De Amescua A., and San T.Feliu, Agile Estimation Techniques and Innovative Approaches to SW Process Improvement. 2014.

50. Combining Challenge-Based Learning and ScrumFramework for Mobile Application Development Alan R. Santos, ITiCSE '15 Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education.

51. Craig Larman Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum/ Craig Larman, Bas Vodde — Kindle Edition, 2015 — 387 p.

52. Da, C. T. F. V., Dantas, V. L. L. Andrade, & R. M. (2011). SLeSS: A scrum and lean six sigma integration approach for the development of sofware customization for mobile phones. Proceedings of the 25th 1263 Brazilian Symposium on Software Engineering (pp. 283-292).

53. David J. S., McCarthy W. E., and Sommer B. S., “Agility – the Key to Survival of the Fittest in the Software Market”, Communication of the ACM, Vol. 46, No. 5, pp.65-69. 2003.

54. Dehlinger J. and Dixon J., “Mobile application SW engineering: Challenges and research directions,” Work. Mob. Softw. Eng., vol. 2, p. 2, 2011.

55. Deloitte, “There’s no place like phone: Consumer usage patterns in the era of peak smartphone,” pp. 1–61, 2016.

56. Dewey J., Experience and Education. Kappa DeltaPi., New York, NY, 1938.].

57. Di Bella, E., Sillitti, A., Succi, G. A multivariate classification of open source developers. Inf. Sciences 221, pp. 72-83. 2013.

58. Doolittle, J., Moohan, A., Simpson, J., Soanes, I.; Building a mobile application development framework. Intel Whitepaper.

59. EBay Enterprise Business Taken Private, Broken Up [Электронный ресурс], URL: https://fortune.com/2015/11/03/ebay-enterprise-sold/ (дата обращения: 09.01.2020).

60. Estrella J. Agile Project Management for Mobile Application Development/ Estrella J. - Kindle Edition, 2017 — 321 p.

61. Flora, H. K., Wang X., and Chande S. V., “An Investigation into Mobile Application Development Processes: Challenges and Best Practices,” Int. J. Mod. Educ. Comput. Sci., vol. 6, no. 6, pp. 1–9, 2014.

62. Green M. Scrum: Novice to Ninja: Methods for Agile, Powerful Development / Green M. - Kindle Edition, 2015 — 278 p.

63. Häkkilä, J., Mäntyjärvi, J.; Developing design guidelines for context-aware mobile applications. 3rd Intl. Conf. on Mobile Technology, Applications and Systems, pp. 1-7. 2006.

64. Hammershoj, A.; Sapuppo, A.; Tadayoni, R.; Challenges for mobile application development. 14th International Conference on Intelligence in Next Generation Networks, pp. 1-8. 2010.

65. Harvard Business Review [Электронный ресурс]. URL: https://hbr.org/2016/04/the-secrethistory-of-agile-innovation (дата обращения: 09.01.2020).

66. Hu, C. and Neamtiu, I., “Automating GUI Testing for Android Applications,” in Proc. of the 6th Int. Workshop on Automation of Software Test, ser. AST ’11, 2011, pp. 77–83.

67. Keith C. Agile Game Development with SCRUM / Keth C. - Addison-Wesley Signature, 2010 — 167 p.

68. Ken Schwaber Agile Project Management with Scrum/ Ken Schwaber — Kindle Edition, 2016 — 589 p.

69. Kniberg H. Scrum and Kanban: making the most of both/ Kniberg H. - Kindle Edition, 2016 — 243 p.

70. Kolb, D. and Fry, R..Toward an Applied Theory ofExperiential Learning. C. Cooper (ed.). Theories ofGroup Process, London: John Wiley., London, 1975.

71. Kylmäkoski, R.; RaPiD7: A collaborative method for the planning activities in software engineering: Industrial experiment. ISBN 952-15-1517-1. Nokia. 2005.

72. La, H.J., Lee, H.J., Kim, S.D.; An Efficiency-centric design methodology for mobile application architectures. 7Th International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 272-279. 2011.

73. McIver, W., “Software Engineering Processes for Mobile Applications Development,” NSERC Mob. First, Frederict., vol. 1, no. 506, pp. 1–74, 2015.

74. Nari Kannan. “Mobile development: Why using an Scrum methodology makes sense”.

75. Norshuhada Shiratuddin, Siti Mahfuzah Sarif: “The md-Matrix: a learning tool in the mobile application development course”, IJMC 7(4): 494-514. 2009.

76. Park, K.; The 4-tier design pattern for the development of an Android Application. LNCS 7105, pp. 196-203. 2011. Springer.

77. Pascoe M. J., “Adding Generic Contextual Capabilities to Wearable Computers,” in Proc. of the 2nd IEEE Int. Symposium on Wearable Computers, ser. ISWC ’98, 1998, pp. 92–95.

78. Paul Duvall Continuous Integration: Improving Software Quality and Reducing Risk / Paul Duvall — Kindle Edition, 2015 — 235 p.

79. Pichler R. Agile product management with Scrum/ Pichler R. - Kindle Edition, 2018 — 298 p.

80. Project Services [Электронный ресурс]. URL: https://www.pmservices.ru/projectmanagement-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/ (дата обращения 09.01.2020).

81. Rahimian V. and Ramsin R., “Designing an Agile Methodology for Mobile SW Development : A Hybrid Method Engineering Approach,” pp. 351–356, 2007.

ДОДАТОК А Графічний матеріал

Плакат 1 Функціонально-логічна структура програмного забезпечення

Плакат 2 Декомпозиція цілей системи

Плакат 3 UML-діаграма варіантів використання

Плакат 4 Схема обробки інформації у програмному забезпеченні

Плакат 5 UML-діаграма класів

Плакат 6 Результати експериментів (1)

Плакат 7 Результати експериментів (2)